Mihaela Malița Mircea Malița

BAZELE INTELIGENȚEI ARTIFICIALE

1 Logici propoziționale

Scanned with OKEN Scanner

MIHAELA MALIȚA

MIRCEA MALIȚA

BAZELE INTELIGENȚEI ARTIFICIALE

1. Logici propoziționale

Chirurtes C.
Iași 5-II-1988

EDITURA TEHNICĂ

București — 1987

Scanned with OKEN Scanner

Primul volum al lucrării este consacrat logicilor propoziționale. Primele cinci capitole sînt o introducere în logica clasică, construirea sistemului ei deductiv, o prezentare a noțiunilor algebrice (latice, algebre pseudobooleene, algebre Boole) necesare pentru înțelegerea și mînuirea modelelor algebrice ale logicilor propoziționale, urmate de algoritmi și circuite logice. Celelalte cinci capitole constituie introduceri în logicile intuiționistă, polivalentă, modală, temporală și probabilistă.

Din punctul de vedere al inteligenței artificiale, acest volum urmărește cunoașterea conceptelor logice de bază, utilizate în procesarea cunoașterii. Volumul este conceput ca o pregătire pentru stăpînirea logicilor predicatelor de ordinul întîi și a demonstrării automate a teoremelor.

Autorii pun accentul pe elaborarea de algoritmi și pe aplicații. Ultimele capitole sînt necesare pentru înțelegerea principalelor tendințe de cercetare în inteligența artificială, unde se utilizează rezultate ale logicilor modale și probabiliste (sistemele expert) și și ale logicii temporale (logica programelor).

Ceea ce distinge această carte de alte lucrări din același domeniu este abordarea paralelă logică și algebrică a calculului propozițiilor. De asemenea, cartea este orientată spre raționamentul și deducția automată a inteligenței artificiale, cu prezentarea unor probleme practice, actuale în cercetare și tehnică.

Lucrarea este adresată informaticienilor din centrele de calcul, inginerilor și proiectanților din industria calculatoarelor și automatică, precum și profesorilor de matematică și informatică din învățămîntul mediu, studenților din facultățile de matematică, informatică, cibernetică, tehnici de calcul etc.

Scanned with OKEN Scanner

PREFAȚĂ

O definiție sugestivă a inteligenței artificiale o prezintă drept acel domeniu de cunoștințe și tehnici, ce permit calculatorului să efectueze operații care, atunci cînd sînt efectuate de om, se numesc inteligente. Această definiție ocolește definiția precisă a inteligenței umane, care a scăpat pînă acum psihologiei, aflată încă în plină dezbatere asupra a ceea ce este inteligența omului și asupra existenței unei singure sau mai multor inteligențe. Operațiile privite drept inteligente la om sînt, printre altele, recunoașterea formelor, recunoașterea vorbirii și, în mod cert, raționamentul. Iată tot atîtea capitole ale inteligenței artificiale, care își propune să elaboreze programe care să facă apt calculatorul de performarea operațiilor amintite. Operațiile fizice constînd din secvențe articulate de mișcări care duc la un anumit scop, ca în producția industrială, sînt inteligente, ceea ce înglobează robotica în inteligența artificială.

Inteligența artificială este un domeniu interdisciplinar. Toate științele care pot concura la modelarea operațiilor inteligente pe calculator sînt prezente: psihologia (în special psihologia învățării), filozofia (în special teoria cunoașterii), matematica cu toate resursele ei (în special logica matematică), fiziologia (în special a creierului), informatica (ca producătoare de programe), electronica și automatica (ca producătoare ale calculatorului). Acestei alianțe de discipline, unite de un singur obiectiv, i se mai spune astăzi știință cognitivă.

În faza cea mai actuală, aceea a elaborării generației a 5-a de calculatoare, cunoștința a fost cosiderată conceptul de bază, în locul informației. Astfel inteligența artificială este uneori privită ca un domeniu al reprezentării cunoștințelor, al procesării lor și al limbajelor artificiale evoluate, care permit reprezentarea și procesarea.

Ca un fir roșu, în toate aceste definiții trece nelipsită inferența logică. Să luăm de exemplu o realizare avansată a inteligenței artificiale, sistemele expert. *Ele sînt formate dintr-o bază de cunoștințe susceptibilă de modificări și de chestionări. În prezența datelor unei probleme, cunoștințele din bază se pronunță și propun soluții. Ca acest lucru să fie posibil, bazei de cunoștințe i se adaugă un sistem de reguli de inferență.*

Istoria efortului de a elucida mecanismul prin care din anumite enunțuri sau propoziții adevărate sau corecte să inferăm alte propoziții adevărate sau corecte începe cu Aristotel și coincide cu istoria logicii. În străduința milenară de a elabora regulile raționamentului corect, în funcție numai de forma și nu de conținutul enunțurilor, ce devin astfel formule, se poate percepe o aspirație spre automatizare. Înainte de a crea mașina electronică de calcul, omul a formalizat mașina deductivă și mașina inferențială în general.

Iată de ce ni se pare justificat să punem printre componentele fundamentale ale inteligenței artificiale logica în expresia ei formalizată și calculatorie, adică logica matematică.

Obiectivul autorilor este prezentarea demonstrării automate a teoremelor, care interesează nu numai matematica, ci și tehnica, un exemplu fiind verificarea automată a programelor și numeroasele aplicații ale inteligenței artificiale. Atingerea acestui obiectiv presupune o introducere în logică. Construcția proiectată are deci două etaje, ultimul fiind demonstrarea

Scanned with OKEN Scanner

automată a teoremelor. Despărțirea logicilor cu predicate (etajul 1) de logicile propoziționale (parterul) s-a născut din examinarea multor manuale și tratate, în care calculul propozițiilor este expediat sumar, toate conceptele și metodele fiind abordate direct pentru calculul cu predicate. În afară de greutatea asimilării logicii predicatelor, această abordare nedreptățește calculul propozițiilor, care este laboratorul logicilor neclasice și al construirii de teorii formalizate.

Explorarea limbajelor de ordinul 0 și a calculelor propoziționale bazate pe ele este obiectul acestui volum. Logica bivalentă, clasică este studiată în primele cinci capitole. O introducere semantică, intuitivă (cap. 1) leagă pe cititor de cunoștințele sale de logică sau permite o rapidă familiarizare cu acestea. Calculul propozițional formalizat (cap. 2) îl introduce în conceptele fundamentale: axiome, reguli de inferență, consistență, completitudine etc. Structura adîncă algebrică a logicii este scoasă apoi în relief (cap. 3), care reprezintă și o introducere în algebra abstractă, căreia îi sînt proprii conceptele de omomorfism, izomorfism etc. Un capitol de algoritmi (cap. 4) fructifică pe planul calculator și practic capitolele anterioare, iar un capitol de introducere în teoria circuitelor logice combinaționale ilustrează aplicațiile algebrelor booleene și ale calculului propozițiilor.

Partea a doua a cărții oferă cititorului elemente de logici neclasice (intuiționistă, polivalentă, modală, temporală și probabilistă). Selecția s-a făcut ținînd seama de preocupările actuale în domeniul inteligenței artificiale de a modela raționamentul asupra incertitudinii. Autorii nu au intenția și nici posibilitatea unei tratări exhaustive: aceasta constituie doar o introducere, un îndemn și o călăuză a cititorilor doritori să le aprofundeze. Preocupările de cercetare și contribuțiile noi ale autorilor sînt reflectate în special în această direcție, mai ales în ultimul capitol.

Orice alegere înseamnă renunțare: nu au putut fi tratate de exemplu rolul grafurilor în logică (în partea întîi), logicile cuantice bazate pe latice nedistributive (în partea a doua). Unele noțiuni sînt dezbătute larg și repetat: ca de exemplu tablele de adevăr, dezvoltarea canonică a formulelor, implicația, pentru că sînt considerate esențiale. Cititorului i se oferă un cîmp de inițiativă pentru demonstrațiile neefectuate în volum, în exerciții și probleme, care se pot face după modelul celor date în volum și pentru investigații proprii, bibliografia oferită fiind concepută ca un ghid în acest scop.

Menționăm că citarea unei cărți la un capitol nu înseamnă că a fost folosită exclusiv la acel capitol.

Cunoștințele de logică și matematică presupuse inițial sînt cele conținute în programa liceului, ca de exemplu noțiunile din teoria mulțimilor, relații, funcții etc.

Am avut în vedere pe utilizatorul posibil al cărții: informaticianul profesionist sau inginerul legat din ce în ce mai mult de calculatoare, pe programatorul doritor să-și completeze cunoștințele în direcția informaticii teoretice, pe profesorul de liceu și pe student, dar ne-am gîndit și la logicienii care vor să studieze algebră și la matematicienii care vor să se informeze în materie de logică.

Cunoștințele din acest volum sînt un prim pas spre înțelegerea unor probleme actuale de inteligență artificială. Următoarele două volume vor aborda problemele privind logicile predicative și demonstrarea automată a teoremelor. Dar cele trei niveluri proiectate nu epuizează cerințele acestei dinamice multidiscipline, noi etaje privind teoria calculabilității, teoria auto-

Scanned with OKEN Scanner

matelor și limbajelor formale, procedeele de reprezentare a cunoașterii fiind susceptibile de construcții ulterioare.

Cuvintele lui Gr. C. Moisil, scrise acum douăzeci de ani, sînt edificatoare pentru viziunea în care această carte a fost concepută și pentru scopurile ei:

„Tehnica nouă: automatica, informatica, cibernetica, lingvistica matematică sînt subsumate unei discipline matematice al cărei rol abia începe să se lămurească. Structura intimă a tuturor acestor capitole ale matematicilor aplicate în tehnica nouă are un caracter algebric".

Mulțumim colegilor C. Calude, A. Iorgulescu, R. Vasile și altora care au formulat observații și propuneri privind redactarea cărții, Editurii Tehnice și redactorului de carte Valentina Crețu pentru interesul, exigența și grija manifestate atît pentru conținutul, cît și pentru forma lucrării.

AUTORII

Scanned with OKEN Scanner

CUPRINS

Listă de notații 8

1. **Logica clasică a propozițiilor** . . 9
   - 1.1. Simboluri. Formule . . . 9
   - 1.2. Evaluare, funcție de adevăr 11
   - 1.3. Tautologie, validitate, consistență 14
   - 1.4. Metoda lui Quine 17
   - 1.5. Metoda reducerii . . . 18
   - 1.6. Echivalențe 19
   - 1.7. Reguli de manipulare și substituție 20
   - 1.8. Dualitate 21
   - 1.9. Discuție despre conectori 22
   - 1.10. Conectori adecvați 23
   - 1.11. Scrierea prefix. Tabloul conectorilor binari . . . 24
   - 1.12. Consecințe logice 26
   - 1.13. Forme normale 28
   - 1.14. Notă istorică 33
   - 1.15. Exerciții și probleme . . 34

   Bibliografie 35

2. **Sistemul deductiv al calculului propozițiilor** 37
   - 2.1. Demonstrație și deducție 37
   - 2.2. Comentarii asupra construirii sistemului L 40
   - 2.3. Sistemele calculului propozițional clasic 42
   - 2.4. Independența axiomelor 44
   - 2.5. Teorema deducției 45
   - 2.6. Consistență, completitudine 47
   - 2.7. Decidabilitate și compactitate 51
   - 2.8. Limbajul formal. Inducție și recurență 53
   - 2.9. Calculul cu arbori de deducție 60
   - 2.10. Teorii formalizate 66
   - 2.11. Notă istorică 68
   - 2.12. Exerciții și probleme . . . 69

   Bibliografie 71

3. **Latice și algebre booleene** . . . 72
   - 3.1. Latice ordonate și latice algebrice 72
   - 3.2. Latice modulare, distributive, complementate . . . 75
   - 3.3. Latice relativ pseudocomplementate și algebre pseudobooleene 76
   - 3.4. Algebre booleene 80
   - 3.5. Relația de congruență și algebre booleene cît . . . 85
   - 3.6. Ideale și filtre 89
   - 3.7. Teoreme de reprezentare. . 92
   - 3.8. Polinoame booleene 93
   - 3.9. Omomorfisme și izomorfisme 95
   - 3.10. Algebre booleene topologice 100
   - 3.11. Algebrele teoriilor propoziționale 102
   - 3.12. Notă istorică 107
   - 3.13. Exerciții și probleme . . . 108

   Bibliografie. 110

4. **Algoritmi** 111
   - 4.1. Algoritmul lui Wang . . . 111
   - 4.2. Regulile Davis-Putnam . . . 113
   - 4.3. Rezoluție 117
   - 4.4. Metodă Veitch-Karnaugh 122
   - 4.5. Algritmul Quine-McCluskey 129
   - 4.6. Simplificare algebrică . . . 131
   - 4.7. Scrierea Galois 133
   - 4.8. Dezvoltări ale funcțiilor booleene 137
   - 4.9. Complexitate 146
   - 4.10. Notă istorică 147
   - 4.11. Exerciții și probleme. . . 148

   Bibliografie. 149

5. **Circuite logice** 150
   - 5.1. Noțiuni introductive . . . 150
   - 5.2. Minimizare 153
   - 5.3. Punți și circuite cu n poli 155
   - 5.4. Modelarea calculului aritmetic prin circuite logice . . . 158
   - 5.5. Proiectarea logică a afișajului 161
   - 5.6. Alte aplicații 162
   - 5.7. Circuite logice combinaționale (CLC) 165
   - 5.8. Notă istorică 170
   - 5.9. Exerciții și probleme . . . 170

   Bibliografie 171

6. **Introducerea în logica intuiționistă și logica pozitivă** 172
   - 6.1. Concepte fundamentale 172
   - 6.2. Logica intuiționistă și algebrele pseudobooleene 173
   - 6.3. Axiomatizări 176
   - 6.4. Logica pozitivă și laticele relativ pseudocomplementate 179
   - 6.5. Notă istorică 181
   - 6.6. Exerciții și probleme . . 181

   Bibliografie 182

7. **Elemente de logică polivalentă** 184
   - 7.1. Construcția semantică . . 184
   - 7.2. Dezvoltarea funcțiilor . . 187
   - 7.3. Algebre Post 191
   - 7.4. Algebra Lukasiewicz-Moisil (LM). 196
   - 7.5 Sisteme logice polivalente 200
   - 7.6. Notă istorică 202
   - 7.7. Exerciții și probleme . . . 203

   Bibliografie 204

Scanned with OKEN Scanner

8. Privire asupra logicilor modale 205

8.1. Operatori modali 205
8.2. Axiome și reguli 208
8.3. Sistemul T 210
8.4. Sistemele S_4 și S_5 212
8.5. Alte sisteme modale . . . 214
8.6. Forma normală conjunctivă modală (FNCM) 216
8.7. Procedura deciziei prin diagrame 218
8.8. Modelele lui Kripke . . . 221
8.9. Logicile modale și algebrele booleene topologice . . . 223
8.10. Logica modală și logica intuiționistă. 224
9.11. Notă istorică 225
8.12. Exerciții și probleme . . . 226
Bibliografie 228

9. Introducere în logica temporală 229

9.1. O logică temporală cu O . 229
9.2. Logica dinamică propozițională 236
9.3. Sistemul R 237
9.4. Logica timpurilor 238
9.5. Logica timpurilor și logicile modale 241
9.6. Notă istorică 241
9.7. Exerciții și probleme . . . 242
Bibliografie 243

10. Elemente de logică probabilistă 244

10.1. Probabilitatea ca măsură a realizabilității 244
10.2. Calculul propozițiilor și calculul probabilităților 247
10.3. Problema lui Boole . . . 248
10.4. Probabilitatea condiționată și condiționalul logic 251
10.5. Inferențe inductive . . . 254
10.6. Logica probabilstă și logica modală 255
10.7. Notă istorică 257
10.8. Exerciții și probleme . . . 258
Bibliografie 258
Index alfabetic 259

Scanned with OKEN Scanner

LISTA DE NOTAȚII

$\triangleq$ reprezintă
$\sim$, $\neg$, $-$, $'$ negație
$\wedge$, $\&$, $\cdot$, și , conjuncție, și
$\vee$ disjuncție, sau
$\rightarrow$, $\supset$ implicație, „dacă ..., atunci ...''
$\leftrightarrow$, $\equiv$ dubla implicație, „dacă și numai dacă''
$\forall$ cuantificatorul universal
$\exists$ cuantificatorul existențial
$\Leftrightarrow$ dubla implicație logică
$\oplus$ suma disjunctivă, suma modulo n
$\overset{*}{\rightarrow}$ conectorul replicație
$\downarrow$ rejecție, conectorul lui Nicod
$\mid$ incompatibilitate, conectorul lui Sheffer
$\triangleright$ relația de dualitate
$\models$ consecința logică, tautologie
$\Rightarrow$ relația de consecință logică
$\vdash$ deducție, teoremă
$\cup$ reuniune, suma
$\cap$ intersecție, produs
$\mathbb{N}$ mulțimea numerelor naturale
$\approx$ relația de echivalență
Δ ideal, operator modal apoi
$-$ diferența de mulțimi
$\cong$ izomorfism
$\Box$ certitudinea, necesarul, clauza vidă
$\Diamond$ posibilul
$\Sigma_{\oplus}$ sumă disjunctivă de termen
$\leftrightarrow$ dubla implicație
$\rightarrow$ implicația modală
S operator modal, S
μ posibilul
γ contingentul
η imposibilul
ν necesarul
$\bigcirc$ următorul
∇ operatorul modal dacă nu
$p(H|E)$ probabilitatea condiționată a lui H
Π probabilul
$\sqcup$ improbabilul
$A/\approx$ mulțimea claselor de echivalență

LISTA DE ABREVIERI

FND forma normală disjunctivă
FNC forma normală conjunctivă
CP calculul propozițiilor
MP modus ponens
Im f imaginea funcției
Ker f nucleul funcției f
P clasa problemelor polinomiale
NP clasa problemelor nepolinomiale
CLC circuit logic combinațional

Scanned with OKEN Scanner

1.
LOGICA CLASICĂ A PROPOZIȚIILOR

1.1. SIMBOLURI. FORMULE

Obiectul logicii este identificarea și folosirea legilor raționamentului uman. Logica simbolică utilizează simboluri pentru unitățile de gîndire și pentru operațiile asupra lor. Următoarele enunțuri sînt propoziții:

$p \triangleq$ Ion citește

$q \triangleq$ Ion scrie

Notația $\triangleq$ arată că propoziția „Ion citește" va fi reprezentată prin simbolul p, iar propoziția „Ion scrie" prin simbolul q. Adresăm unei propoziții p o singură cerință: să fie adevărată sau falsă, fără să poată fi adevărată sau falsă în același timp. Pentru a spune că nu este adevărat că „Ion citește" avem nevoie de un semn nou, negația reprezentată prin $\sim$:

$\sim p \triangleq$ Nu este adevărat că Ion citește (Ion nu citește).

Atîta timp cît nu admitem decît aceste două cazuri: p și $\sim p$ sau două valori de adevăr (fals, adevărat) sîntem în cadrul *logicii clasice* sau *logicii bivalente*. Ridicarea acestei restricții și acceptarea unui număr mai mare decît două valori de adevăr ne va situa într-o *logică neclasică* (*polivalentă*).

Propoziția logică diferă de propoziția gramaticală, care, deși corectă, poate să nu fie neapărat adevărată sau falsă. De exemplu, propoziția imperativă: „Citește, Ion!" sau „Mîine, Ion va scrie".

În logica de tip propozițional, propozițiile sînt unități nedecompozabile. Orice considerație asupra structurii lor interne este irelevantă. Spre deosebire de îmbrățișarea acestui postulat, logica predicatelor reprezintă enunțurile sub formă de predicate. De exemplu, „Ion este harnic" se scrie cu simbolul $H(\text{Ion})$ și mai general $H(x)$, unde $x = \text{Ion}$ din mulțimea persoanelor $\{x\}$ care satisfac predicatul H sau are proprietatea H (a fi harnic). Logica predicatelor cere noi simboluri de operații asupra enunțului $H(x)$ care sînt *cuantificatorii* $\forall$, $(\forall x)$: pentru orice x și $\exists$, $(\exists x)$: pentru unii x. În cele ce urmează ne vom ocupa în exclusivitate de logica propozițiilor și nu vom utiliza simbolurile specifice logicii predicatelor.

Propozițiile simple sau *atomi* $p, q, r, \ldots$ sînt compozabile. Ele se pot combina, dînd naștere la noi propoziții, la rîndul lor adevărate sau false. Simbolurile numite *conectori* indică operațiile de asociere sau combinare, care sînt cele mai frecvente în vorbire sau raționament. Ele sînt următoarele:

$\wedge$ pentru „și": $p \wedge q \triangleq$ „Ion citește" și „Ion scrie"

Scanned with OKEN Scanner

$\vee$ pentru „sau": $p \vee q \triangleq$ „Ion citește" sau „Ion scrie"

$\rightarrow$ pentru „dacă p, atunci q": $p \rightarrow q \triangleq$ Dacă „Ion citește", atunci „Ion scrie"

$\leftrightarrow$ pentru „p dacă și numai dacă q": $p \leftrightarrow q$

Propozițiile $p \wedge q$, $p \vee q$, $p \rightarrow q$ sînt formule propoziționale. Formulele vor fi notate cu majuscule: P, Q, R, ... De exemplu:

$$P : (p \wedge q) \rightarrow r,$$

unde $r \triangleq$ Ion este mulțumit. Interpretarea formulei P este: dacă „Ion citește" și „Ion scrie", atunci „Ion este mulțumit". Formula P este formată din trei atomi. Ea poate fi privită drept o funcție de atomii p, q, r, care, din acest motiv, se numesc și *variabile propoziționale*. O întrebare la care trebuie să știm să răspundem în logica propozițiilor este următoarea: care este valoarea de adevăr a formulei P (adevărat sau fals), atunci cînd cunoaștem valorile de adevăr ale variabilelor propoziționale p, q, r, ... ce o compun?

Abordarea semantică a capitolului 1 presupune că răspunsul la această întrebare face apel la evaluări și interpretări (cercetarea valorii de adevăr a formulelor). Capitolul următor ne va situa în cadrul logicii formale, care se ocupă de studiul validității în sensul de corectitudine a unei formule, numai pe baza formei sau sintaxei sale. În logica formală a propozițiilor nu vom recurge la interpretări. Simbolurile p, q, r, ... și operațiile asupra lor vor fi abstracții, susceptibile desigur de interpretarea ca propoziții, dar aceasta ulterior și nu în cadrul demonstrației și calculului. Atita timp cît acceptăm referirea la faptul că p, q, r, ... sînt propoziții (și eventual le exemplificăm), spunem că lucrăm „intuitiv", în contrast cu stilul „formal".

În logica propozițiilor, unitatea principală de lucru este *formula*. Dacă formulele ar avea numai două sau trei variabile propoziționale și unul sau doi conectori logici, am putea să ne pronunțăm asupra valorii de adevăr a formulelor fără ajutorul unui aparat simbolic. Dar formulele asupra cărora sîntem chemați să ne pronunțăm, pe care le transformăm și simplificăm, la care stabilim echivalențe sau consecințe logice, au uneori zeci de variabile propoziționale, legate de un număr de conectori. Logica propozițiilor urmărește stăpînirea acestei complexități, prin elaborarea unui calcul adecvat.

Totalitatea simbolurilor logicii propozițiilor constituie un *alfabet*. În cele ce urmează *simbolurile logice* vor fi următoarele:

$$p,\ q,\ r,\ \ldots,\ \sim,\ \wedge,\ \vee,\ \rightarrow,\ \leftrightarrow,\ (,)$$

cu alte cuvinte, simbolurile pentru mulțimea finită sau infinită de atomi: p, q, r, ... sau p_1, p_2, ..., p_n, ...; simbolurile pentru conectorii logici, ce vor fi definiți mai jos; parantezele, stîngă, (, și dreaptă,).

Simbolurile ce nu sînt cuprinse în alfabet sînt *simboluri metalogice*. Această restricție surprinde de la început: simbolurile P, Q, R, ... sau P_1, P_2, P_3, ... utilizate pentru formule care nu figurează în alfabet.

Alfabetul împreună cu formulele formează *limbajul*. Proprietățile, legile, regulile pot fi enunțate pentru atomi și atunci avem nevoie de o regulă suplimentară de trecere de la atomi la formule (regula substituției) sau pot fi enunțate direct pentru formule, caz în care poartă denumirea de *scheme*.

Scanned with OKEN Scanner

Exemplu: $p \wedge (p \to q) \to q$ este o formulă întotdeauna adevărată (tautologie), oricare ar fi atomii p și q. Iar $P \wedge (P \to Q) \to Q$, unde P, Q sînt formule, este și ea întotdeauna adevărată, oricare ar fi P, Q, de exemplu $P = (p \leftrightarrow q)$ și $Q = (p \vee q) \wedge (p \vee \sim q)$.

Legea tautologică enunțată pentru atomi poate fi generalizată prin *regula substituției*. Este convenabil să o enunțăm ca schemă în forma generală pentru formule, fără a mai menționa substituția.

O formulă P se obține din atomii p, q, r, ... prin aplicarea operațiilor efectuate de conectori. Aceștia transformă atomii după cum urmează:

conectorul $\sim$ transformă pe p în $\sim p$,
conectorul $\wedge$ transformă perechea p, q în $p \wedge q$,
conectorul $\vee$ transformă perechea p, q în $p \vee q$,
conectorul $\to$ transformă perechea p, q în $p \to q$,
conectorul $\leftrightarrow$ transformă perechea p, q în $p \leftrightarrow q$.

Conectorul $\sim$ se aplică la un singur atom (sau formulă) și de aceea se numește *unar*, iar ceilalți patru operează asupra a doi atomi, purtînd numele de *conectori binari*.

Definiția recursivă a *formulei* este:

1) un atom este o formulă;
2) dacă P este o formulă, $(\sim P)$ este o formulă;
3) dacă P și Q sînt formule, atunci $(P \wedge Q)$, $(P \vee Q)$, $(P \to Q)$, $(P \leftrightarrow Q)$ sînt formule;
4) toate formulele se obțin prin aplicarea regulilor 1) —3).

Observăm că o formulă se scrie riguros începînd cu o paranteză stîngă și terminîndu-se cu o paranteză dreaptă. În uzul curent, omitem aceste paranteze ori de cîte ori absența lor nu dă naștere la echivoc.

Exemple. 1) $(P \to \ )$ nu este o formulă. Implicația nu leagă două formule.
2) $(P \to (Q \wedge ((\sim R) \vee S)))$ este o formulă.

1.2. EVALUARE, FUNCȚIE DE ADEVĂR

Introducem pe mulțimea atomilor o funcție, numită *evaluare*, care aplică fiecare atom pe mulțimea (adevărat, fals). Identificăm falsul cu 0, adevărul cu 1. Pentru fiecare atom p vom avea $v(p) = 0$ sau $v(p) = 1$. O logică care are două valori de adevăr (adevărat și fals) se numește *logică bivalentă*.

Prima problemă a calculului propozițiilor constă în determinarea valorii de adevăr a unei formule în funcție de valoarea de adevăr a atomilor. Notăm valoarea de adevăr a unei formule P cu $P^v \in \{0, 1\}$, sau cu notația echivalentă $v(P)$.

Stabilim prin recurență următoarele reguli pentru evaluarea formulelor:

1) dacă P este un atom, $P^v = v(P)$;

2) dacă $P = \sim Q$, atunci $P^v = \begin{cases} 1, & \text{dacă } Q^v = 0, \\ 0, & \text{dacă } Q^v = 1; \end{cases}$

3) $(P \wedge Q)^v = \begin{cases} 1, & \text{dacă } P^v = Q^v = 1, \\ 0 & \text{altfel;} \end{cases}$

Scanned with OKEN Scanner

4) $(P \vee Q)^v = \begin{cases} 0, \text{ dacă } P^v = Q^v = 0, \\ 1 \text{ altfel;} \end{cases}$

5) $(P \to Q)^v = \begin{cases} 0, \text{ dacă } P^v = 1 \text{ și } Q^v = 0, \\ 1 \text{ altfel;} \end{cases}$

6) $(P \leftrightarrow Q)^v = \begin{cases} 1, \text{ dacă } P^v = Q^v, \\ 0 \text{ altfel.} \end{cases}$

Aceste convenții pot fi enunțate în felul următor :

a) Negația lui P, notată cu $\sim P$ (scrisă uneori $\neg P$ sau $\bar{P}$), este adevărată cînd P este falsă și este falsă cînd P este adevărată (se citește „non-P").

b) Conjuncția $P \wedge Q$ este adevărată cînd P și Q sînt adevărate, altfel este falsă (se citește „P și Q").

c) Disjuncția $P \vee Q$ este adevărată cînd cel puțin una din P și Q este adevărată, altfel este falsă (se citește „P sau Q").

d) Implicația $P \to Q$ este falsă cînd P este adevărată și Q este falsă (se citește „dacă P, atunci Q" sau „P implică Q").

e) Dubla implicație $P \leftrightarrow Q$ este adevărată cînd P și Q au aceleași valori de adevăr, altfel este falsă (se citește „dacă P, atunci și numai atunci Q").

Regulile 1) — 6) pot fi descrise în tabloul următor :

P^v	Q^v	$\sim P^v$	$(P \wedge Q)^v$	$(P \vee Q)^v$	$(P \to Q)^v$	$(P \leftrightarrow Q)^v$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

De obicei, în loc de P^v, Q^v, $\sim P^v$, $(P \wedge Q)^v$, ... deasupra coloanelor tablei de adevăr se scrie P, Q, $\sim P$, $(P \wedge Q)$, ... cu înțelegerea că este vorba de valoarea de adevăr a acestor formule.

Un tablou care arată valoarea de adevăr a unei formule pentru toate valorile posibile ale atomilor din formulă se numește *tablă de adevăr*. Tabloul de mai sus cuprinde cinci table de adevăr ale formulelor $\sim P$, $P \wedge Q$, $P \vee Q$, $P \to Q$, $P \leftrightarrow Q$.

Funcția de adevăr face să corespundă în mod unic fiecărei evaluări a atomilor unei formule una din valorile de adevăr 0 sau 1. Pentru negație funcția de adevăr este $f^{\sim}(1) = 0$, $f^{\sim}(0) = 1$. Reprezentarea grafică a unei funcții de adevăr este tabla de adevăr. În cazul conjuncției, funcția de adevăr f este reprezentată de tabla de adevăr :

	P^v	Q^v	$(P \wedge Q)^v$
$f^{\wedge}(1, 1) = 1$	1	1	1
$f^{\wedge}(1, 0) = 0$	1	0	0
$f^{\wedge}(0, 1) = 0$	0	1	0
$f^{\wedge}(0, 0) = 0$	0	0	0

Pe coloana întîi apar evaluările formulei P, pe coloana a doua evaluările lui Q. Cele patru linii ale primelor două coloane sînt evaluările perechii P, Q. În ultima coloană apar evaluările lui $P \wedge Q$. Întregul tablou este

Scanned with OKEN Scanner

reprezentarea grafică a funcției de adevăr notată $f^{\wedge}(P, Q)$ a formulei $P \wedge Q$.

Unei formule cu n atomi, concepuți ca variabile propoziționale, îi va corespunde o funcție de adevăr de n variabile. Tabla de adevăr va avea 2^n linii pentru cele 2^n evaluări posibile.

Există 2^{2^n} funcții de adevăr distincte pentru o formulă cu n variabile, deoarece există 2^{2^n} moduri de aranjare a lui 0 și 1 în ultima coloană a unei table de adevăr cu 2^n linii. În cazul $n = 2$ al formulelor cu doi atomi există 16 formule de adevăr. Se pot astfel defini 16 conectori binari, din care am selectat doar patru.

Numărul formulelor care pot fi construite cu n variabile propoziționale este infinit. Rezultă că diferite formule corespund uneia și aceleiași funcții de adevăr (table de adevăr).

Exemple. 1) Să se scrie tabla de adevăr a formulei

$$(P \wedge Q) \rightarrow (R \leftrightarrow (\sim S)).$$

La întocmirea unei table de adevăr este esențial să se epuizeze totalitatea combinațiilor de valori de adevăr (evaluările) ale atomilor. În prima coloană succesiunea de 1 și 0 va cuprinde două grupuri, a doua coloană — patru și ș.a.m.d. În acest caz tabla va avea $2^4 = 16$ linii, după cum se vede din tabelul 1.1.

Tabelul 1.1

P	Q	R	S	$\sim S$	$P \wedge Q$	$R \leftrightarrow (\sim S)$	$(P \wedge Q) \rightarrow (R \leftrightarrow (\sim S))$
1	1	1	1	0	1	0	0
1	1	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	0	0	1	1	0	0
1	0	1	1	0	0	0	1
1	0	1	0	1	0	1	1
1	0	0	1	0	0	1	1
1	0	0	0	1	0	0	1
0	1	1	1	0	0	0	1
0	1	1	0	1	0	1	1
0	1	0	1	0	0	1	1
0	1	0	0	1	0	0	1
0	0	1	1	0	0	0	1
0	0	1	0	1	0	1	1
0	0	0	1	0	0	1	1
0	0	0	0	1	0	0	1

2) Este mai simplu și mai economic să scriem tablele de adevăr, scriind formula pe linia de sus a tabloului, evaluările atomilor pe coloana în dreptul fiecărei apariții a lor și apoi în dreptul conectorilor valoarea de adevăr a rezultatului acțiunii lor.

Fie formula $(p \wedge q) \rightarrow p$. Avem :

$(p$	$\wedge$	$q)$	$\rightarrow$	p
1		1		1
1		0		1
0		1		0
0		0		0

Completăm apoi coloana de sub $\wedge$:

$(p$	$\wedge$	$q)$	$\rightarrow$	p
1	1	1		1
1	0	0		1
0	0	1		0
0	0	0		0

Scanned with OKEN Scanner

Şi, în sfîrşit, sub implicaţia care leagă coloana a doua de ultima:

$(p$	$\wedge$	$q)$	$\rightarrow$	p
1	1	1	1	1
1	0	0	1	1
0	0	1	1	0
0	0	0	1	0

3) Tabla de adevăr a formulei $\sim(p \wedge q) \leftrightarrow ((\sim p) \vee (\sim q))$ în această scriere este

$\sim$	$(p$	$\wedge$	$q)$	$\leftrightarrow$	$((\sim p)$	$\vee$	$(\sim q))$
0	1	1	1	1	0	0	0
1	1	0	0	1	0	1	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

Interpretare, model. Pînă acum o tablă de adevăr a fost considerată ca o funcţie de adevăr definită de ultima coloană, în timp ce coloanele ce corespund atomilor cuprind toate evaluările posibile (combinaţiile de 0 şi 1).

Definiţii. Fie o formulă P avînd atomii $p_1, \ldots, p_n$. O *interpretare* a lui P este o evaluare (o atribuire de valori) a atomilor, fiecare primind valoarea de adevăr 0 sau 1. Formula P este adevărată într-o interpretare, dacă este evaluată ca 1; altfel este falsă în interpretare. O formulă cu n atomi va avea 2^n interpretări diferite. Liniile unei table de adevăr reprezintă astfel de interpretări.

O interpretare I se reprezintă prin mulţimea $a_1, \ldots, a_n$, unde a_i este p_i sau $\sim p_i$. Astfel $\{p, \sim q, \sim r, s\}$ este o interpretare, în care p, q, r, s au respectiv evaluările 1, 0, 0, 1. Dacă P este adevărată în interpretarea I, se spune că I *satisface pe* P sau P *este satisfăcut* de I. Dacă P este falsă în interpretarea I, atunci I *falsifică pe* P sau P *este falsificat de* I. O interpretare I care satisface pe P se numeşte *model al lui* P.

Exemplu. Fie formula $P: \sim(p \vee q) \vee \sim(q \vee p)$, cu tabla de adevăr

p	q	$p \vee q$	$\sim(p \vee q)$	$\sim(q \vee p)$	$\sim(p \vee q) \vee \sim(q \vee p)$
1	1	1	0	0	0
1	0	1	0	0	0
0	1	1	0	0	0
0	0	0	1	1	1

Formula P este satisfăcută de interpretarea $\{\sim p, \sim q\}$ care constituie un model al lui P, întrucît evaluările $p = 0$, $q = 0$ le corespunde $P = 1$. Interpretarea $\{p, q\}$ falsifică pe P.

1.3. TAUTOLOGIE, VALIDITATE, CONSISTENŢĂ

Definiţii. O formulă este *validă*, dacă şi numai dacă este adevărată în toate interpretările. Altfel (dacă şi numai dacă nu este validă) este *nevalidă*.

O formulă este *inconsistentă* dacă şi numai dacă este falsă în toate interpretările. O formulă este *consistentă* dacă şi numai dacă nu este inconsistentă.

Scanned with OKEN Scanner

Denumirea clasică a formulei valide din punct de vedere semantic este *tautologie*, iar a celei inconsistente este *contradicție*.

Exemple. 1) Formula $P : p \vee \sim p$ este o tautologie (formulă validă), întrucît funcția sa de adevăr ia în toate interpretările valoarea 1 :

p	$\sim p$	$p \vee \sim p$
1	0	1
0	1	1

2) Formula $Q : p \wedge \sim p$ este o contradicție (formulă nevalidă) pentru că funcția sa de adevăr ia în toate interpretările valoarea 0 :

p	p	$p \wedge \sim p$
1	0	0
0	1	0

Definițiile date justifică următoarele

Propoziții. 1) *O formulă este validă dacă și numai dacă negația ei este inconsistentă*

2) *O formulă este inconsistentă dacă și numai dacă negația ei este validă.*

3) *O formulă este nevalidă dacă și numai dacă există o inerpretare în care este falsă.*

4) *O formulă este consistentă dacă și numai dacă există o interpretare în care este adevărată.*

5) *O formulă validă este consistentă.*

6) *O formulă consistentă nu este neapărat validă.*

7) *O formulă inconsistentă este nevalidă.*

8) *O formută nevalidă nu este neapărat inconsistentă.*

Exemple. Formula $P : p \vee \sim p$ fiind validă este consistentă. Formula $Q : p \wedge \sim p$ fiind inconsistentă este nevalidă. Formula $R : p \rightarrow \sim p$ cu tabla de adevăr

p	$\sim p$	$p \rightarrow \sim p$
1	0	0
0	1	1

este nevalidă, dar consistentă.

Întrucît folosirea terminologiei de mai înainte depășește necesitățile ce se ivesc în demonstrarea automată a teoremelor, sînt folosite cu precădere următoarele expresii :

O formulă consistentă se numește *realizabilă*. O formulă este *nerealizabilă* (inconsistentă) dacă și numai dacă nu este realizabilă.

Definițiile proprietăților de validitate (tautologii), consistență (realizabile), inconsistență (contradicții, nerealizabile) și nevaliditate a formulelor sînt reprezentate de tabelul 1.2.

Tabelul 1.2

CONSISTENTĂ = REALIZABILĂ $P^v = 1$ uneori	NEVALIDĂ $P^v = 0$ uneori
VALIDĂ = TAUTOLOGIE $P^v = 1$ întotdeauna	CONTRADICȚIE = NEREALIZABILĂ $P^v = 0$ întotdeauna

Proprietatea de formulă nerealizabilă devine centrală, dată fiind abordarea caracteristică demonstrării automate a teoremelor care folosește cu precădere *principiul contradicției*.

O formulă este nerealizabilă cînd este falsă în toate interpretările, deci cînd nu admite model. Dacă reușim să demonstrăm că există cel puțin o interpretare pentru care este adevărată (admite cel puțin un model), ipoteza nu este adevărată și formula se dovedește a fi realizabilă.

Proprietatea poate fi enunțată pentru o mulțime $\mathcal{S}$ de formule: o astfel de mulțime este realizabilă (ca funcție de adevăr), dacă există o evaluare v pe mulțimea formulelor astfel ca $P^v = 1$ pentru toți $P \in \mathcal{S}$; altfel $\mathcal{S}$ este nerealizabilă. Dacă $\mathcal{S}$ constă dintr-o singură formulă, se spune că aceasta este *realizabilă*, dacă $\mathcal{S}$ este realizabilă, și *nerealizabilă*, dacă $\mathcal{S}$ este nerealizabilă.

În limbajul evaluărilor o tautologie este o formulă T pentru care $T^v = 1$, oricare ar fi evaluarea v.

Tautologii fundamentale. Următoarele formule valide sînt folosite cu precădere, ceea ce le-a făcut să se distingă uneori prin nume proprii. Ele sînt tautologii:

T0	$P \to P$	
T1	$(P \to Q) \to ((Q \to R) \to (P \to R))$	legea silogismului,
T2	$P \to (P \vee Q)$	adunare,
T3	$Q \to (P \vee Q)$	
T4	$(P \to R) \to ((Q \to R) \to ((P \vee Q) \to R))$	
T5	$P \wedge Q \to P$	
T6	$P \wedge Q \to Q$	
T7	$(R \to P) \to ((R \to Q) \to (R \to (P \wedge Q)))$	
T8	$(P \to (Q \to R)) \to ((P \wedge Q) \to R)$	legea importării,
T9	$((P \wedge Q) \to R) \to (P \to (Q \to R))$	legea exportării,
T10	$(P \wedge \sim P) \to Q$	
T11	$(P \to (P \wedge \sim P)) \to \sim P$	
T12	$P \wedge \sim P$	legea terțului exclus sau tertium non datur

Sînt frecvente și următoarele legi:

T13	$(P \wedge (P \to Q)) \to Q$	modus ponendo ponens sau **modus ponens** sau regula detașării,
T14	$(\sim Q \wedge (P \to Q)) \to \sim P$	modus tollendo tollens
T15	$(\sim P \wedge (P \vee Q)) \to Q$	modus tollendo ponens,
T16	$((P \to Q) \wedge \sim Q) \to \sim P$	absurditate,
T17	$\sim(P \wedge \sim P)$	negare (contradicție),
T18	$P \to (Q \to P)$	simplificare,
T19	$(P \to (Q \to R)) \to (Q \to (P \to R))$	schimbarea premiselor,
T20	$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$	legea lui Frege,
T21	$P \to \sim \sim P$	
T22	$\sim \sim P \to P$	
T23	$(P \to Q) \to (\sim P \vee Q)$	
T24	$(\sim P \vee Q) \to (P \to Q)$	
T25	$\sim(P \vee Q) \to (\sim P \wedge \sim Q)$	T25 — T28: legile lui
T26	$(\sim P \wedge \sim Q) \to \sim(P \vee Q)$	De Morgan
T27	$\sim(P \wedge Q) \to (\sim P \vee \sim Q)$	
T28	$(\sim P \vee \sim Q) \to \sim(P \wedge Q)$	

Scanned with OKEN Scanner

T29	$(P \to Q) \to (\sim Q \to \sim P)$	T29 – T32 : legile transpunerii
T30	$(P \to \sim Q) \to (Q \to \sim P)$	
T31	$(\sim P \to Q) \to (\sim Q \to P)$	
T32	$(\sim P \to \sim Q) \to (Q \to P)$	
T33	$((P \to Q) \to P) \to P$	legea lui Peirce.

Demonstrația caracterului tautologic al formulelor T0 — T33 se face prin scrierea tablei de adevăr a fiecărei formule, a cărei ultimă coloană va fi compusă din 1.

1.4. METODA LUI QUINE

Un procedeu de a decide asupra realizabilității unei formule (sau a caracterului de tautologie sau contradicție) care nu recurge la tabla de adevăr a fost dat de Quine. S-a urmărit evitarea tablei de adevăr, care devine greu de utilizat pentru un număr mare de atomi. Ea constă în următorii pași :

1) Atomul care intervine cel mai frecvent este înlocuit o dată cu V (adevăr) și a doua oară cu F (fals), obținîndu-se două formule.

2) În fiecare din ele se aplică următoarele reguli :

a) Se suprimă V ca element al conjuncției.
b) Se suprimă F ca element al disjuncției.
c) O conjuncție cu F se înlocuiește cu F.
d) O disjuncție cu V se înlocuiește cu V.
e) O implicație conținînd pe V se înlocuiește cu consecventul.
f) O implicație conținînd pe F se înlocuiește cu negația antecedentului.
g) Se suprimă V ca membru al unei duble negații.
h) Se înlocuiește dubla negație, care conține F ca membru, cu celălalt membru al ei.

3) Se aplică regula 1) la următorul atom cu frecvența cea mai mare și se trece la regula 2), apoi la regula 3), pînă la epuizarea atomilor.

Toate regulile se bazează pe tablele de adevăr ce definesc conectorii. Astfel regula b), $F \vee q = q$, exprimă faptul că dacă p este fals, $p \vee q$ ia valoarea de adevăr a lui q.

Regula d), $V \vee q = V$, se reduce la faptul că o disjuncție este adevărată dacă un element este adevărat.

Regula f), cu privire la $F \to q$, sau $p \to F$, se deduce din

p	q	$p \to q$
1	1	1
1	0	0
0	1	1
0	0	1

În primul caz $F \to q$ înseamnă că $p \to q$ este adevărată, deci putem s-o înlocuim cu negația antecedentului (V). În al doilea caz, $p \to F$, evaluările (1, 0) și (0, 0), rezultă din tablou că are întotdeauna valoarea lui $\sim p$.

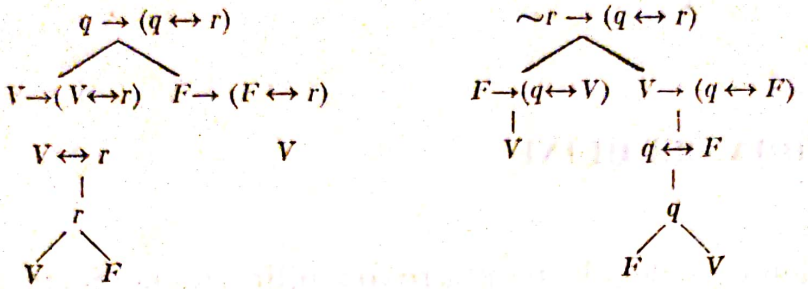
Exemplu. Fie formula $(p \wedge q) \vee (\sim p \wedge \sim r) \rightarrow (q \leftrightarrow r)$. Este o tautologie sau o contradicție? Aplicăm metoda lui Quine:

$$((p \wedge q) \vee (\sim p \wedge \sim r)) \rightarrow (q \leftrightarrow r),$$

$$((V \wedge q) \vee (F \wedge \sim r)) \rightarrow (q \leftrightarrow r), \quad ((F \wedge q) \vee (V \wedge \sim r)) \rightarrow (q \leftrightarrow r),$$

$$(q \vee (F \wedge \sim r)) \rightarrow (q \leftrightarrow r), \quad (F \vee (V \wedge \sim r)) \rightarrow (q \leftrightarrow r),$$

$$(q \vee F) \rightarrow (q \leftrightarrow r), \quad (V \wedge \sim r) \rightarrow (q \leftrightarrow r),$$



Formula nu este tautologie sau contradicție, ea este realizabilă.

1.5. METODA REDUCERII

Metoda reducerii se bazează pe încercarea de a falsifica formula dată, al cărei caracter vrem să-l demonstrăm. Presupunem că există o evaluare v pentru care $F^v = 0$. Înscriem această valoare (0) sub conectorul principal, iar deasupra lui numărul pasului efectuat (1). Dacă formula este de forma $P \rightarrow Q$, pasul 1 va consta în

$$P \overset{1}{\underset{0}{\rightarrow}} Q.$$

Dar implicația $P \rightarrow Q$ este falsă numai cînd P este fals și Q este adevărat. Vom scrie deci sub conectorul principal din P valoarea 1, sub conectorul principal din Q valoarea 0 și deasupra ambilor conectori numărul pasului (2). Continuăm în același fel pînă cînd (a) obținem o evaluare consistentă (necontradictorie) pentru toate variabilele propoziționale sau (b) nu găsim o asemenea evaluare. În primul caz, ipoteza inițială este corectă și formula este falsă; în al doilea caz falsificarea nu a reușit și deci formula este validă.

Cînd formula este de forma $P \overset{1}{\underset{0}{\wedge}} Q$, din atribuirea valorii 0 nu rezultă în mod unic că P este adevărat sau fals, de aceea trebuie să considerăm toate cazurile:

2	1	2	2	1	2	2	1	2
P	$\wedge$	Q	P	$\wedge$	Q	P	$\wedge$	Q
1	0	0	0	0	1	0	0	0

unde valorile și indicii se pun sub și deasupra conectorilor principali din formulele P și Q.

Dacă fiecare din cele trei cazuri conduc la contradicții, formula inițială este validă. În caz că cel puțin unul din ele permite găsirea unei evaluări necontradictorii, formula a putut fi falsificată și este inconsistentă. Evaluările pentru care se obține o contradicție sînt subliniate.

Scanned with OKEN Scanner

Exemple. 1) Fie formula $((p \to q) \wedge r) \to ((\sim r \vee p) \to q)$. Aplicind metoda reducerii, se obține o contradicție prin faptul că la pasul 3 : $v(r) = 1$, iar la pasul 9 : $v(r) = 0$

$$\begin{array}{ccccccccccccc} 6 & 3 & 5 & 2 & 3 & 1 & & 8 & 9 & 4 & 7 & 2 & 4 \\ ((p & \to & q) & \wedge & r) & \to & & ((\sim r & \vee & p) & \to & q) & \\ 0 & 1 & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & 0 & 0 & 0 \end{array}$$

Observăm că la pasul 3 din faptul că valoarea conjuncției din membru intii este 1, rezultă că $v(p \to q) = 1$ și $v(r) = 1$. Întrucit $v(q) = 0$ (rezultă din membrul doi la pasul 4), $v(p) = 0$ (altfel $v(p \to q) = 0$). Dar dacă introducem $v(p) = 0$ (din membrul intii, pasul 6) în membrul doi (pasul 7), obținem $v(\sim r) = 1$ și $v(r) = 0$, ceea ce contrazice valoarea care i s-a atribuit lui r la pasul 3. Formula este o tautologie.

2) Fie formula $((\sim r \vee p) \to q) \to ((p \to q) \wedge r)$. Întrucit la pasul 3 valoarea lui $\sim r \vee p$ nu rezultă în mod unic, trebuie să considerăm următoarele trei cazuri :

$$\text{a)}\ \begin{array}{ccccccc} 3 & & 2 & 3 & 1 & & 2 \\ ((\sim r & \vee & p) & \to & q) & \to & ((p \to q) \wedge r) \\ 1 & & 1 & 1 & 0 & & 0 \end{array}$$

$$\text{b)}\ \begin{array}{ccccccc} 3 & & 2 & 3 & 1 & & 2 \\ ((\sim r & \vee & p) & \to & q) & \to & ((p \to q) \wedge r) \\ 0 & & 1 & 1 & 0 & & 0 \end{array}$$

$$\text{c)}\ \begin{array}{ccccccc} 3 & & 2 & 3 & 1 & & 2 \\ ((\sim r & \vee & p) & \to & q) & \to & ((p \to q) \wedge r). \\ 0 & & 1 & 0 & 0 & & 0 \end{array}$$

Cazul b) și cazul c) duc la contradicții, dar cazul a) permite o evaluare consistentă :

$$\begin{array}{ccccccccccccc} ((\sim & r & \vee & p) & \to & q) & \to & ((p & \to & q) & \wedge & r) \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}$$

$v(p) = 1$, $v(q) = 1$, $v(r) = 0$. Formula nu este o tautologie. Ea este falsificată de interpretarea $\{p, q, \sim r\}$. Formula este realizabilă, pentru că alte interpretări, ca de exemplu $\{p, q, r\}$, contrazic ipoteza nerealizabilității ei.

1.6. ECHIVALENȚE

Definiție. Două formule sînt (logic) *echivalente* (P echivalent cu Q) dacă iau aceleași valori de adevăr pentru orice interpretare. Aceasta revine la faptul că au aceeași tablă de adevăr. Scriem $P \Leftrightarrow Q$ dacă pentru orice evaluare v avem $P^v = Q^v$.

O definiție echivalentă este : două formule P și Q sînt echivalente, dacă $P \leftrightarrow Q$ este o tautologie.

Echivalențe fundamentale. Următoarele echivalențe, demonstrate cu ajutorul tablelor de adevăr, sînt frecvent utilizate, sub denumiri proprii :

1) Absorbția $P \wedge (P \vee Q) \Leftrightarrow P$ și $P \vee (P \wedge Q) \Leftrightarrow P$.

2) Idempotența $P \wedge P \Leftrightarrow P$ și $P \vee P \Leftrightarrow P$.

3) Comutativitatea $P \wedge Q \Leftrightarrow Q \wedge P$ și $P \vee Q \Leftrightarrow Q \vee P$.

4) Asociativitatea $P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$,

$$P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R.$$

5) Distributivitatea $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$,

$$P \vee (Q \wedge R) \Rightarrow (P \vee Q) \wedge (P \vee R).$$

Scanned with OKEN Scanner

6) Dubla negație $\sim P(\sim P) \Leftrightarrow P$.

7) Legile lui De Morgan $\sim(P \wedge Q) \Leftrightarrow \sim P \vee \sim Q$, $\sim(P \vee Q) \Leftrightarrow \sim P \wedge \sim Q$.

8) Transpunerea $P \to Q \Leftrightarrow \sim Q \to \sim P$.

9) Negația implicației $\sim(P \to Q) \Leftrightarrow P \wedge \sim Q$.

10) Formula bicondițională $P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\sim P \wedge \sim Q)$.

1.7. REGULI DE MANIPULARE ȘI SUBSTITUȚIE

Dacă P și $P \to Q$ sînt tautologii, atunci Q este o tautologie.

Presupunem că P și $P \to Q$ sînt tautologii, dar Q nu este. Atunci există o evaluare a atomilor ce apar în P și Q, care dau lui Q valoarea 0. Dar P este o tautologie, $P = 1$. Aceasta dă lui $P \to Q$ valoarea 0, ceea ce contrazice ipoteza că $P \to Q$ este tautologie (vezi T13). Vom nota această regulă MP de la *modus ponens*.

Fie F o formulă în care apar atomii $p_1, \ldots, p_n$. Fie $P_1, \ldots, P_n$ formule oarecare. Dacă F este o tautologie, atunci $\mathscr{F}$ obținută prin substituirea peste tot în F a fiecărei apariții a lui p_i prin P_i $(1 \leqslant i \leqslant n)$ este de asemenea o tautologie.

Demonstrația se bazează pe faptul următor: dacă atribuim valori de adevăr atomilor ce apar în $P_1, \ldots, P_n$, valoarea luată de $\mathscr{F}$ este una și aceeași cu valoarea pe care o ia F cînd $p_1, \ldots, p_n$ au valorile asumate de $P_1, \ldots, P_n$, adică 1 întrucît F este o tautologie. Deci și $\mathscr{F}$ este o tautologie.

Exemple. 1) Prin tabla de adevăr se demonstrează că formula următoare este o tautologie:

$$\sim(p \wedge q) \leftrightarrow \sim p \vee \sim q \text{ (De Morgan).}$$

Înlocuind atomii p, q cu formulele P, Q, avem tautologia

$$\sim(P \wedge Q) \leftrightarrow \sim P \vee \sim Q,$$

unde P și Q sînt formule oarecare.

2) Întrucît $p_1 \wedge (p_2 \wedge p_3) \leftrightarrow (p_1 \wedge p_2) \wedge p_3$ este o tautologie (asociativitatea), oricare ar fi formulele P, Q, R,

$$P \wedge (Q \wedge R) \leftrightarrow (P \wedge Q) \wedge R \text{ va fi o tautologie.}$$

Regula 4) din 1.6 ne-a permis să scriem cele zece echivalențe direct cu litere mari (pentru formule), deși demonstrația se face în prealabil pe tablele de adevăr, în care variabilele sînt scrise cu litere mici (atomi). Folosim evident definiția unei echivalențe ca o dublă implicație tautologică.

Într-o formulă $\mathscr{P}$ în care intervine subformula P, înlocuim pe P în unul sau mai multe locuri cu formula echivalentă Q. Rezultatul $\mathscr{Q}$ este logic echivalent cu $\mathscr{P}$. Să arătăm că $\mathscr{P} \leftrightarrow \mathscr{Q}$ este o tautologie. Fie o evaluare a atomilor. Formula $\mathscr{P}$ are aceeași valoare ca și Q, deoarece diferă de aceasta numai prin faptul că P este înlocuit cu Q avînd aceeași valoare de adevăr. Deci $\mathscr{P} \leftrightarrow \mathscr{Q}$ ia întotdeauna valoarea 1.

Exemplu. Formula $(p \wedge p) \to q$ este echivalentă cu $p \to q$, deoarece am făcut substituția $p \wedge p \Leftrightarrow p$, care reprezintă o echivalență logică.

Prin folosirea semnului egalității toate tautologiile sînt echivalente și toate formulele nerealizabile sînt echivalente. Întrucît $P^\nu = 1$ pentru toți

Scanned with OKEN Scanner

v și $Q^v = 0$ pentru toți v determină pe P ca o tautologie și pe Q ca o formulă nerealizabilă, putem scrie 1 și 0 pentru orice formulă care este tautologie, respectiv nerealizabilă. De aceea $P = 1$ înseamnă că P este tautologie și $Q = 0$ că Q este o formulă nerealizabilă.

1.8. DUALITATE

O *formulă restrînsă* se numește o formulă care nu are decît conectorii $\sim$, $\wedge$, $\vee$.

Dacă într-o formulă restrînsă P schimbăm între ele simbolurile $\wedge$ și $\vee$ și înlocuim fiecare atom prin negația lui, obținem formula P^*. Atunci P^* este logic echivalent cu $\sim P$: $P^* \Leftrightarrow \sim P$.

Această proprietate, numită și *principiul dualității*, rezultă din faptul că în tabla de adevăr a conectorilor $\wedge$ și $\vee$ valorile se obțin una din alta prin schimbarea între ele a lui 0 și 1.:

p	q	$p \wedge q$	$p \vee q$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Întrucit demonstrația riguroasă se face prin inducție referitoare la numărul de conectori, o vom prezenta detaliat. Va fi suficient să dovedim că o formulă cu n conectori satisface enunțul dualității.

Fie $n = 0$. În acest caz P nu are conectori și se reduce la un atom $P \Leftrightarrow p$. Înlocuind pe p prin $\sim p$, primim $P^* \Leftrightarrow \sim p \Leftrightarrow \sim P$, ceea ce corespunde proprietății de dualitate. Pentru $n > 0$ (P are n conectori) presupunem că dualitatea este valabilă la oricare formulă cu un număr mai mic de n conectori.

Cazul 1. P este de forma $\sim Q$. Deci Q are $n - 1$ conectori, deci prin ipoteză $Q^* \Leftrightarrow \sim Q$. Dar $P^* \Leftrightarrow \sim Q^*$, deci $P^* \Leftrightarrow \sim(\sim Q) \Rightarrow \sim P$.

Cazul 2. P este de forma $Q \vee R$, unde Q și R au mai puțin de n conectori. Atunci $Q^* \Leftrightarrow \sim Q$ și $R^* \Leftrightarrow \sim R$. Dar

$$P^* \Leftrightarrow Q^* \wedge R^* \Leftrightarrow (\sim Q) \wedge (\sim R) \Leftrightarrow \sim(Q \vee R) \Leftrightarrow \sim P.$$

Cazul 3. P este de forma $Q \wedge R$, unde Q și R au mai puțin de n conectori. Atunci $Q^* \Leftrightarrow \sim Q$, $R^* \Leftrightarrow \sim R$, iar

$$P^* \Leftrightarrow Q^* \vee R^* = (\sim Q) \vee (\sim R) \Leftrightarrow \sim(Q \wedge R) \Leftrightarrow \sim P.$$

S-a arătat astfel că, în ipoteza că principiul dualității se aplică la formule cu n conectori, acesta este valabil pentru orice formulă cu n conectori.

Principiul general al dualității afirmă că *un enunț corect implicînd $\wedge$, $\vee$, 0, 1 poate fi tradus într-un alt enunț corect prin schimbările între ele ale simbolurilor 0 și 1, $\wedge$ și $\vee$*. Noțiunile definite în termenii acestor simboluri trebuie traduse prin dualele lor. Astfel dualul termenului „tautologie" este „nerealizabil" pentru că cele două enunțuri duale sînt:

$P^v = 1$ pentru toți v și $P^v = 0$ pentru toți v.

Astfel dualul enunțului „dacă P este o tautologie, $P \vee Q$ este o tautologie" este „dacă P este nerealizabil, $P \wedge Q$ este nerealizabil".

1.9. DISCUȚIE DESPRE CONECTORI

Calculul propozițiilor încearcă să formalizeze înlănțuirea corectă a propozițiilor din limbajul curent. Cei cinci conectori folosiți surprind expresii gramaticale simple care leagă propozițiile în uzul lor obișnuit. În ceea ce privește negația și conjuncția, acestea pot fi citite în mod neambiguu.

Exemplu. Fie $p\underline{\Delta}$ afară plouă, $q\underline{\Delta}$ afară este frig. (Vom reprezenta în continuare prin $\underline{\Delta}$ corespondența între un atom și o propoziție.) Negația lui p este $\sim p\underline{\Delta}$ afară nu plouă, sau $\sim p\underline{\Delta}$ nu este adevărat că afară plouă. Astfel $p \wedge q$ va fi în mod evident propoziția : „afară plouă și afară este frig".

În ceea ce privește disjuncția intervine o ambiguitate, $p \vee q$ este propoziția „afară plouă **sau** afară este frig". Dar în limbajul obișnuit sau înseamnă fie una din cele două situații, prin excluderea celeilalte (ori/ori), fie cel puțin una, dar poate și amîndouă. De aceea în logică se face distincție între *disjuncția simplă* și *disjuncția exclusivă* $\oplus$ (suma disjunctivă) comparate în următoarele table de adevăr :

p	q	$p \vee q$	$p \oplus q$
1	1	1	0
1	0	1	1
0	1	1	1
0	0	0	0

Grigore Moisil a citit disjuncția simplă $\vee$ prin **vel** (p sau măcar q). Disjuncția exclusivă în care $p \oplus q$ înseamnă că una din propoziții este adevărată și cealaltă falsă este citită **aut**.

Legătura dintre ele este dată de relațiile

$$p \oplus q = (p \wedge \sim q) \vee (q \wedge \sim p),$$

$$p \oplus q = (p \vee q) \wedge (\sim p \vee \sim q).$$

Exemple. „Aut Cesar, aut nihil". „Ușa este închisă **sau** deschisă". *Vel* și *aut* sînt expresiile latine pentru **sau**.

Implicație este conectorul asupra căruia în istoria logicii s-au purtat cele mai vii discuții : $p \to q$ se citește : „dacă p, atunci q", „p implică q", „p numai dacă q", „q presupunînd p", „p este condiția suficientă a lui q" și „q este condiția necesară a lui p". Prin faptul că implicația este falsă într-o singură situație și anume cînd p este adevărată și q falsă, se produce o îndepărtare de la uzul curent al vorbirii.

Cazurile mult discutate sînt :

a) în definirea implicației falsul implică orice, conform enunțului latin *falso sequitur quodlibet* ;

b) adevărul este implicat de orice : *verum sequitur ad quodlibet*. Faptul că implicația este adevărată cînd p este fals, iar q este adevărat sau fals nu se împacă cu vorbirea curentă.

Scanned with OKEN Scanner

Exemple. „Dacă $1 + 1 = 2$, Parisul este capitala Franței", „Dacă luna este din brînză, atunci $2 \times 2 = 4$".

Pentru remedierea situației s-a propus conectorul *replicație*, notat cu simbolul $\overset{*}{\rightarrow}$ și a cărui tablă de adevăr comparată cu a implicației este următoarea :

p	q	$p \rightarrow q$	$p \overset{*}{\rightarrow} q$
1	1	1	1
1	0	0	1
0	1	1	0
0	0	1	1

Se observă că prin replicație se elimină „falso sequitur quodlibet". Întrucît implicția stă la baza relației de consecință logică, definiția este decisivă pentru structura unei logici.

1.10. CONECTORI ADECVAȚI

O mulțime de conectori se numește *mulțime adecvată*, dacă fiecare formulă poate fi exprimată cu ajutorul conectorilor acestei mulțimi.

S-a văzut pînă acum că $\{\sim, \wedge, \vee\}$ este o mulțime adecvată de conectori.
Perechile $\{\sim, \wedge\}$, $\{\sim, \vee\}$ și $\{\sim, \rightarrow\}$ sînt și ele mulțimi adecvate de conectori.

Folosim formulele lui De Morgan pentru a înlocui orice formulă de forma $P \vee Q$ prin $\sim(\sim P \wedge \sim Q)$ și astfel am eliminat disjuncția. La fel se elimină conjuncția prin formulele $P \wedge Q = \sim(\sim P \vee \sim Q)$ și astfel mulțimea $\{\sim, \vee\}$ se dovedește a fi adecvată. Pentru a treia mulțime facem apel la $P \wedge Q = \sim(P \rightarrow \sim Q)$ și $P \vee Q = \sim P \rightarrow Q$.

Nici o altă pereche de conectori din mulțimea $\{\sim, \wedge, \vee, \rightarrow\}$ nu este adecvată. Mulțimea $\{\sim, \leftrightarrow\}$ nu este adecvată. Întrebarea „Există mulțimi adecvate formate dintr-un singur conector?" conduce în mod surprinzător la un răspuns pozitiv. Pentru aceasta introducem conectorul $\downarrow$ numit *rejecție* sau *conectorul lui Nicod*, care se citește „nici ..., nici ...". În literatura tehnică este cunoscut sub denumirea de NOR. Tabla sa de adevăr este următoarea :

p	q	$p \downarrow q$
1	1	0
1	0	0
0	1	0
0	0	1

Relațiile care îl leagă de negație și conjuncție sînt :

$$\sim p = p \downarrow p \quad \text{și} \quad p \wedge q = (p \downarrow p) \downarrow (q \downarrow q).$$

Conectorul numit *incompatibilitate* sau *conectorul lui Sheffer*, numit NAND, este definit de relația $p \mid q = \sim p \vee \sim q$, ceea ce înseamnă că este adevărat cînd ambii atomi nu sînt adevărați :

p	q	$p \mid q$
1	1	0
1	0	1
0	1	1
0	0	1

Scanned with OKEN Scanner

Relaţiile pertinente sînt:

$$\sim p = p|p \text{ şi } p \wedge q = (p|p) \mid (q|q).$$

Din faptul că $\sim$, $\wedge$ şi $\sim$, $\vee$ se exprimă în funcţie de conectorul lui Nicod, respectiv Sheffer, iar aceste perechi formează mulţimi adecvate, rezultă că mulţimile $\{\downarrow\}$ şi $\{|\}$ sînt adecvate.

Exemplu. Vom scrie $p \to q$ folosind conectorul lui Nicod:

$$p \to q = \sim p \vee q = \sim (p \wedge \sim q) = \sim (p \wedge (q \downarrow q)) =$$

$$= \sim ((p \downarrow p) \downarrow ((q \downarrow q) \downarrow (q \downarrow q))) =$$

$$= ((p \downarrow p) \downarrow ((q \downarrow q) \downarrow (q \downarrow q))) \downarrow ((p \downarrow p) \downarrow ((q \downarrow q) \downarrow (q \downarrow q))).$$

Avantajul de a micşora numărul de conectori este plătit prin lungimea scrierii formulelor.

1.11. SCRIEREA PREFIX. TABLOUL CONECTORILOR BINARI

Iniţial ne-am servit de cinci conectori, din care patru binari: conjuncţia, disjuncţia, implicaţia şi dubla implicaţie. Am făcut apoi cunoştinţă cu disjuncţia exclusivă, replicaţia, conectorii lui Nicod şi Sheffer. Numărul de conectori binari este 16, reprezentînd numărul coloanelor de 4 elemente cu 0 şi 1, pe care le putem scrie pentru a defini tablele de adevăr ale acestor conectori.

Am utilizat pînă acum scrierea „infix" în care conectorul este scris între propoziţiile asupra cărora operează. Există diferenţe de notaţie între şcolile principale de logică. Astfel Russell utilizează pentru negaţie simbolul $\sim p$, iar Hilbert $\bar{p}$.

Scrierea „prefix" sau „scrierea poloneză" iniţiată de Lukasiewicz pune conectorul la începutul perechii de propoziţii pe care le leagă. Dăm mai jos un tablou comparativ al scrierii conectorilor:

În lucrarea de faţă	Peano-Russell	Hilbert	Lukasiewicz
$\sim p$	$\sim p$	$\bar{p}$	Np
$p \wedge q$	$p \cdot q$	$p\&q$	Kpq
$p \vee q$	$p \vee q$	$p \vee q$	Apq
$p \to q$	$p \supset q$	$p \to q$	Cpq
$p \leftrightarrow q$	$p \equiv q$	$p \sim q$	Epq

Exemple. a) Formula $(p \vee q) \wedge (q \to r)$ se scrie la Peano-Russell $p \vee q \cdot q \supset r$ şi la Lukasiewicz $KApqCqr$.

b) Formula $(p \to q) \to (\sim q \to \sim p)$ se scrie la Peano-Russell $p \supset q \cdot \supset \cdot \sim q \supset \sim p$ şi la Lukasiewicz $CCpqCNqNp$.

Tabelul 1.3 reprezintă cei 16 conectori binari posibili identificaţi după scrierea „prefix".

Scanned with OKEN Scanner

Tabelul 1.3

p / *q*	1 1	1 0	0 1	0 0	Infix	Nume
Vpq	1	1	1	1		Tautologia
Opq	0	0	0	0		Contradicția (Interferența)
Epq	1	0	0	1	$\leftrightarrow$	Echivalența (Implicația dublă)
Jpq	0	1	1	0	$\oplus$	Disjuncția exclusivă, sumă disjunctivă
Fpq	0	0	1	1		Prenonpendența
Gpq	0	1	0	1		Inhibiția
Hpq	1	0	1	0		Postpendența
Ipq	1	1	0	0		
Apq	1	1	1	0	$\vee$	Disjuncția, OR
Kpq	1	0	0	0	$\wedge$	Conjuncția, AND
Bpq	1	1	0	1		Replicația (Implicația materială inversă)
Lpq	0	1	0	0		Excepția
Cpq	1	0	1	1	$\rightarrow$	Implicația materială
Mpq	0	0	1	0		Negația implicației materiale
Dpq	0	1	1	1	$\mid$	Incompatibilitatea. Excluzia. Conectorul lui Sheffer, NAND
Xpq	0	0	0	1	$\downarrow$	Rejecția. Conectorul lui Nicod (Peirce), NOR

Conectorii din tabelul 1.3 sînt legați de relații de negație și dualitate (pe care o vom nota cu $\triangleright$):

$$NVpq = Opq, \quad NEpq = Jpq, \quad NFpq = Ipq,$$
$$NHpq = Gpq, \quad NKpq = Dpq, \quad NApq = Xpq,$$
$$NBpq = Mpq, \quad NCpq = Lpq,$$
$$\triangleright Vpq = Opq, \quad \triangleright Eqp = Jpq, \quad \triangleright Apq = Kpq,$$
$$\triangleright Bpq = Lpq, \quad \triangleright Cpq = Mpq.$$

Conectorii F, G, H, I sînt autoduali. Relațiile între conectori sînt arătate în pătratul lui Boetius (fig. 1.1).

Pătratul lui Boetius (fig. 1.2), care a fost construit pentru propozițiile universal afirmative A (toți da), universal negative E (toți nu), particular afirmative I (unii da), particular negative O (unii nu), este caracterizat de următoarele proprietăți:

Subalternare: din adevărul supraalternant rezultă adevărul subalternant, nu și invers; din falsul subalternant rezultă falsul supraalternant, nu și invers.

Scanned with OKEN Scanner

Contradicție: nu pot fi împreună adevărate sau împreună false.
Subcontrarietate: nu pot fi ambele false, dar pot fi împreună adevărate.

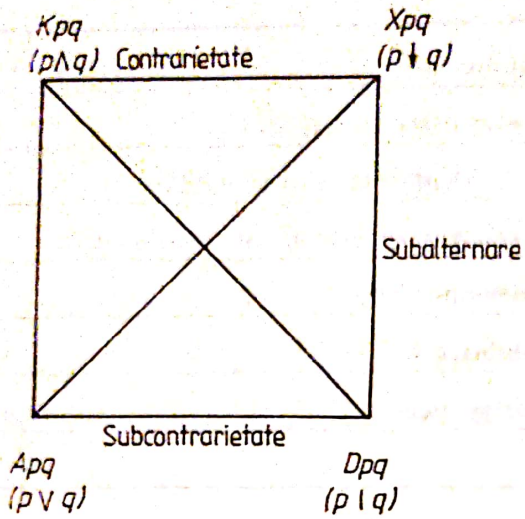


Fig. 1.1.

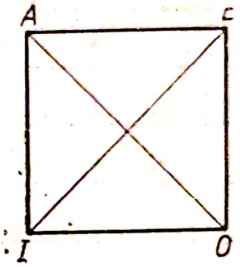


Fig. 1.2.

Relațiile de negație și dualitate dintre conectori justifică faptul că un număr limitat dintre ei servește necesitățile calculului propozițiilor.

1.12. CONSECINȚE LOGICE

Fie n formule $P_1, P_2, \ldots, P_n$. Formula P este o *consecință logică* (tautologică) a premiselor $P_1, \ldots, P_n$ dacă pentru fiecare evaluare v pentru care $P_1^v = \ldots = P_n^v = 1$ avem $P^v = 1$.

Folosim pentru consecința logică notația $P_1, \ldots, P_n \models P$. Simbolul $\models$ în fața unei formule denotă că este o tautologie: $\models P$ (citim P este tautologie).

Cu alte cuvinte, orice model al conjuncției premiselor $P_1, P_2, \ldots, P_n$ este și model al concluziei.

Se demonstrează următoarele propoziții:

Propoziția 1. *Fie formulele $P_1, P_2, \ldots, P_n$. Formula P este o consecință logică a premiselor P_i, dacă și numai dacă $P_1 \wedge \ldots \wedge P_n \wedge \sim P$ este nerealizabilă.*

Demonstrație. Presupunem că $P_1 \wedge \ldots \wedge P_n \to P$ nu ar fi o tautologie. În această ipoteză există o evaluare v, pentru care $(P_1 \wedge \ldots \wedge P_n) \to P$ nu ar fi o tautologie. În această ipoteză există o evaluare v, pentru care $(P_1 \wedge P_2 \wedge \ldots \wedge P_n)^v = 1$ și $P^v = 0$, sau $P_1^v = \ldots = P_n^v = 1$ și $P^v = 0$, ceea ce contrazice definiția pentru $P_1, P_2, \ldots, P_n \models P$.

Pentru teorema a doua, ipoteza că $P_1 \wedge P_2 \wedge \ldots \wedge P_n \wedge \sim P$ ar fi realizabilă conduce de asemenea la contrazicerea definiției, pentru că ar însemna că $P_1^v = P_2^v = \ldots = P_n^v = \sim P^v = 1$ sau $P_1^v = \ldots = P_n^v = 1$ și $P^v = 0$.

Problema stabilirii dacă o formulă este consecință logică (numită și *inferență tautologică*) revine astfel la stabilirea faptului dacă o altă formulă este tautologie sau nerealizabilă. Pentru un număr redus de variabile propoziționale se poate recurge la tabla de adevăr a acestei formule.

Scanned with OKEN Scanner

Exemple. Fie $P_1 \underset{\sim}{\Delta} A \to B$, $P_2 \underset{\sim}{\Delta} \sim B$, $P \underset{\sim}{\Delta} \sim A$. Să se demonstreze că $P_1, P_2 \models P$.

1)

A	B	$A \to B$	$\sim B$	$(A \to B) \wedge \sim B$	$\sim A$
1	1	1	0	0	0
1	0	0	1	0	0
0	1	1	0	0	1
0	0	1	1	1	1

Conform definiției, evaluarea (0, 0), care dă valoarea 1 conjuncției premiselor, face pe P adevărată.

2)

A	B	$A \to B$	$\sim B$	$(A \to B) \wedge \sim B$	$\sim A$	$((A \to B) \wedge \sim B) \to \sim A$
1	1	1	0	0	0	1
1	0	0	1	0	0	1
0	1	1	0	0	1	1
0	0	1	1	1	1	1

Am aplicat propoziția 1 : $P_1 \wedge P_2 \to P$ este o tautologie.

3)

A	B	$A \to B$	$\sim B$	$(A \to B) \wedge \sim B$	$(A \to B) \wedge \sim B \wedge A$
1	1	1	0	0	0
1	0	0	1	0	0
0	1	1	0	0	0
0	0	1	1	1	0

Am aplicat propoziția 2 : $P_1 \wedge P_2 \wedge \sim P$ este nerealizabilă.

Cînd numărul de atomi ai formulei este mare, recurgerea la tabla de adevăr nu este practicabilă.

Exemplu. Fie $((p \wedge q) \to (r \to s))$, $(p_1 \wedge q_1 \to r_1)$, $(r_1 \wedge s \to s_1)$, p, q, q_1, $p_1 \models s_1$. Sînt 8 atomi, ceea ce ar da un tablou cu $2^8 = 256$ linii. Un raționament direct permite demonstrația. Fie v o evaluare pentru care premisele sînt adevărate. Atunci $(p \wedge q)^v = (p_1 \wedge q_1)^v = 1$. De aici $(r \wedge s)^v = r_1^v = 1$ și $s^v = 1$. Deci $(r_1 \wedge s)^v = 1$ și $s_1^v = 1$.

Introducem notația $\Rightarrow$ pentru relația de consecință logică definită în propoziția 1. Folosirea acestei notații aduce simplificări în scriere și raționament. În loc de a spune că $P_1, \ldots, P_n \to P$ este o tautologie, scriem $P_1, \ldots, P_n \Rightarrow P$.

În loc de a spune $T_2 : P_1 \to P \vee Q$ este o tautologie scriem $T_2 : P_1 \Rightarrow P \vee Q$. Este de asemenea valabilă scrierea $T_2 : \models P_1 \to P \vee Q$.

Conectorul $\to$ este o **operație** din care rezultă o nouă formulă $R : P \to Q$, în timp ce $P \Rightarrow Q$ indică doar o **legătură** între formulele P și Q, care conform definiției se rezumă la „P nu poate fi adevărat și Q fals", întrucît $P \to Q$ este întotdeauna adevărată.

Să presupunem că $n = 1$, iar $P_n = p$, $P = q$. În acest caz are loc relația dintre atomii $p \Rightarrow q$. În tabla de adevăr a implicației

p	q	$p \to q$	$p \Rightarrow q$	$p \vee q$	$p \wedge q$
1	1	1	1	1	1
1	0	0	—	—	—
0	1	1	1	1	0
0	0	1	1	0	0

excludem linia a doua. Atunci se vede că $p \vee q \Leftrightarrow q$ și $p \wedge q \Leftrightarrow p$.

Scanned with OKEN Scanner

Putem enunța:

$$p \Rightarrow q \text{ dacă și numai dacă } p \vee q \Leftrightarrow q \text{ și } p \wedge q \Leftrightarrow p.$$

Observăm că $p \vee q \Leftrightarrow q$ și $p \wedge q \Leftrightarrow q$ se deduc una din cealaltă. Într-adevăr, dacă $p \vee q \Leftrightarrow q$, atunci $p \wedge (p \vee q) \Leftrightarrow p \wedge q$. Dar membrul întîi este p, conform legii absorbției, deci $p \Leftrightarrow p \wedge q$.

Invers, dacă $p \wedge q \Leftrightarrow p$, folosim echivalența

$$q \Leftrightarrow (p \wedge q) \vee (\sim p \wedge q) \Leftrightarrow p \vee (\sim p \wedge q) \Leftrightarrow p \vee q.$$

Dacă $p \Rightarrow q$, $(p \to q)^v = 1$, $(\sim p \vee q)^v = 1$ sau, folosind legile lui De Morgan,

$$(\sim(p \wedge \sim q))^v = 1 \text{ sau } (p \wedge \sim q)^v = 0.$$

Din formula $p \Leftrightarrow (p \wedge q) \vee (p \wedge \sim q)$ deducem

$$p^v = (p \wedge q)^v + (p \wedge \sim q)^v = (p \wedge q)^v$$

și deci $p \Leftrightarrow p \wedge q$. Reciproc, din $p \Leftrightarrow p \wedge q$ rezultă $p^v = (p \wedge q)^v$ și deci $(p \wedge \sim q)^v = 0$ sau $(\sim p \vee q)^v = 1$ și $(p \to q)^v = 1$ sau $p \Rightarrow q$.

Relația $\Rightarrow$ este o relație de preordine (cvasiordine) întrucît este caracterizată de proprietățile reflexivității și tranzitivității:

1) $P \Rightarrow P$ (vezi T0 din 1.3).

2) Dacă $P \Rightarrow Q$, $Q \Rightarrow P$, atunci $P \Leftrightarrow Q$.

Într-adevăr, dacă $P \to Q$, $Q \to P$ sînt tautologii, $P \leftrightarrow Q$ este tautologie și, conform definiției din 1.6, avem $P \Leftrightarrow Q$, dar nu și $P = Q$ (antisimetrie).

3) $P \Rightarrow Q$, $Q \Rightarrow R$ atrage $P \Rightarrow R$. Se demonstrează cu ajutorul T1 din 1.3.

Relația de preordine $\Rightarrow$ induce relația de echivalență $\Leftrightarrow$, conform cu 2), care este reflexivă, simetrică și tranzitivă. Aceasta explică notația $\Leftrightarrow$ folosită pentru echivalență. Rezumăm notațiile și propozițiile lor prin tabloul:

Conectori (operații)		Relații (operațiile sînt tautologice)	
Implicația	$\to$	Relația de consecință logică	$\Rightarrow$
Dubla implicație	$\leftrightarrow$	Echivalența	$\Leftrightarrow$

În timp ce $\to$, $\leftrightarrow$ sînt simboluri logice cuprinse în alfabet, $\Rightarrow$ și $\Leftrightarrow$ sînt simboluri metalogice.

1.13. FORME NORMALE

Legea asociativității ne permite să ne debarasăm de paranteze în expresiile legate numai de disjuncție:

$$P_1 \vee (P_2 \vee P_3) = (P_1 \vee P_2) \vee P_3 = P_1 \vee P_2 \vee P_3.$$

În general, se scrie $\bigvee_{i \le n} P_i = P_1 \vee P_2 \cdots \vee P_n$ pentru disjuncția formulelor P_i, care prezintă proprietatea de a fi adevărată dacă cel puțin una este adevărată.

Scanned with OKEN Scanner

Analog se scrie conjuncția formulelor P_i, care este adevărată dacă și numai dacă toate P_i sînt adevărate:

$$\bigwedge_{i \leq n} P_i = P_1 \wedge P_2 \wedge \ldots \wedge P_n.$$

Definiții. Un *literal* este un atom sau negația unui atom. O formulă P este în *forma normală conjunctivă* (*FNC*) care are forma $P_1 \wedge P_2 \wedge \ldots \wedge P_n$, unde P_i este o disjuncție de literale.

O formulă P este în *forma normală disjunctivă* (*FND*) dacă are forma $P_1 \vee P_2 \vee \ldots \vee P_n$, unde P_i este o conjuncție de literale.

Exemple. Formula $(p \vee q \vee \sim r) \wedge (p \vee \sim q \vee r) \wedge (\sim p \vee \sim q \vee r)$ este în forma normală conjunctivă (*FNC*).

Formula $(p \wedge \sim q) \vee (\sim p \wedge \sim q) \vee (p \wedge q)$ este o formă normală disjunctivă (*FND*).

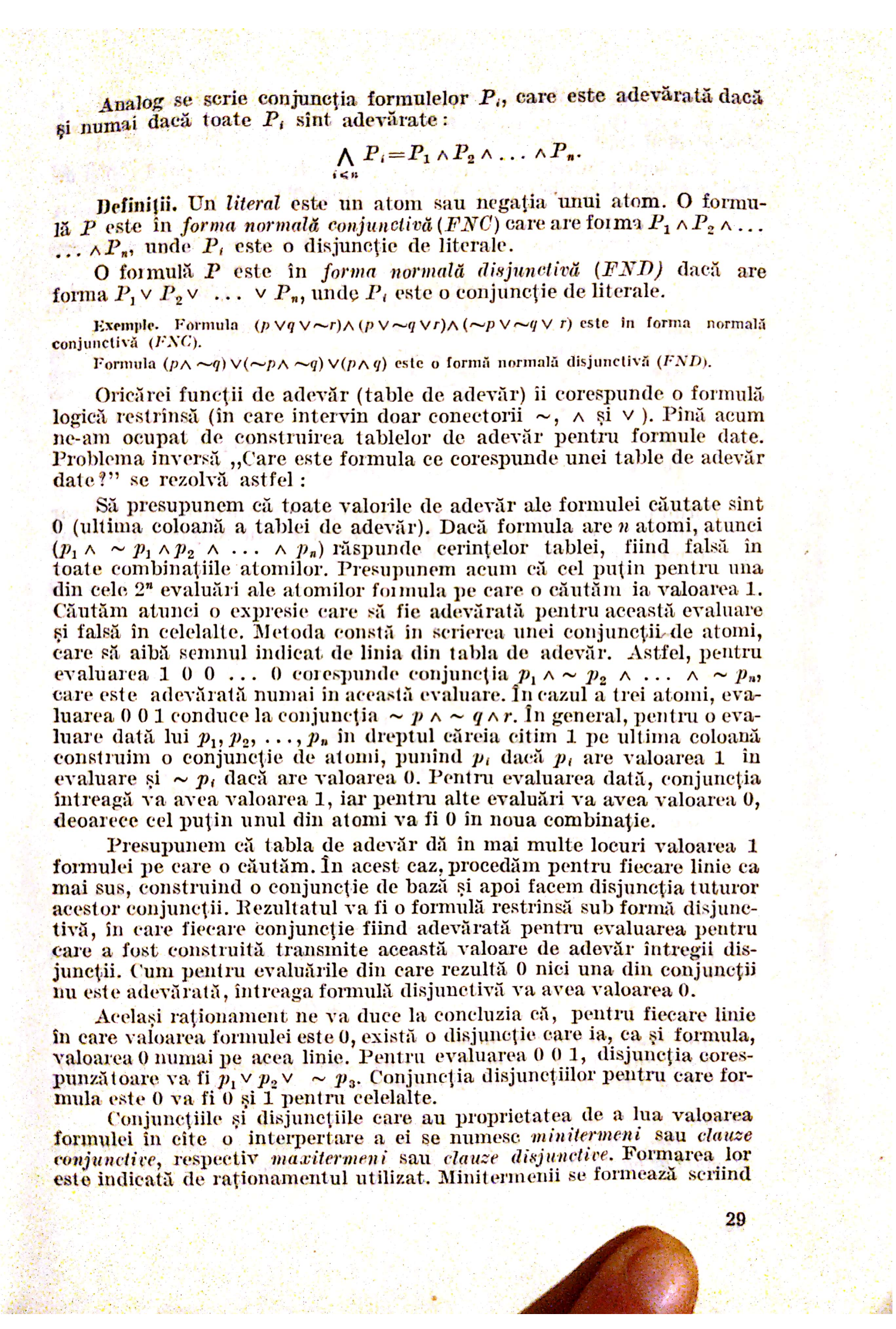
Oricărei funcții de adevăr (table de adevăr) îi corespunde o formulă logică restrînsă (în care intervin doar conectorii $\sim$, $\wedge$ și $\vee$). Pînă acum ne-am ocupat de construirea tablelor de adevăr pentru formule date. Problema inversă „Care este formula ce corespunde unei table de adevăr date?" se rezolvă astfel:

Să presupunem că toate valorile de adevăr ale formulei căutate sînt 0 (ultima coloană a tablei de adevăr). Dacă formula are n atomi, atunci $(p_1 \wedge \sim p_1 \wedge p_2 \wedge \ldots \wedge p_n)$ răspunde cerințelor tablei, fiind falsă în toate combinațiile atomilor. Presupunem acum că cel puțin pentru una din cele 2^n evaluări ale atomilor formula pe care o căutăm ia valoarea 1. Căutăm atunci o expresie care să fie adevărată pentru această evaluare și falsă în celelalte. Metoda constă în scrierea unei conjuncții de atomi, care să aibă semnul indicat de linia din tabla de adevăr. Astfel, pentru evaluarea 1 0 0 ... 0 corespunde conjuncția $p_1 \wedge \sim p_2 \wedge \ldots \wedge \sim p_n$, care este adevărată numai în această evaluare. În cazul a trei atomi, evaluarea 0 0 1 conduce la conjuncția $\sim p \wedge \sim q \wedge r$. În general, pentru o evaluare dată lui $p_1, p_2, \ldots, p_n$ în dreptul căreia citim 1 pe ultima coloană construim o conjuncție de atomi, punînd p_i dacă p_i are valoarea 1 în evaluare și $\sim p_i$ dacă are valoarea 0. Pentru evaluarea dată, conjuncția întreagă va avea valoarea 1, iar pentru alte evaluări va avea valoarea 0, deoarece cel puțin unul din atomi va fi 0 în noua combinație.

Presupunem că tabla de adevăr dă în mai multe locuri valoarea 1 formulei pe care o căutăm. În acest caz, procedăm pentru fiecare linie ca mai sus, construind o conjuncție de bază și apoi facem disjuncția tuturor acestor conjuncții. Rezultatul va fi o formulă restrînsă sub formă disjunctivă, în care fiecare conjuncție fiind adevărată pentru evaluarea pentru care a fost construită transmite această valoare de adevăr întregii disjuncții. Cum pentru evaluările din care rezultă 0 nici una din conjuncții nu este adevărată, întreaga formulă disjunctivă va avea valoarea 0.

Același raționament ne va duce la concluzia că, pentru fiecare linie în care valoarea formulei este 0, există o disjuncție care ia, ca și formula, valoarea 0 numai pe acea linie. Pentru evaluarea 0 0 1, disjuncția corespunzătoare va fi $p_1 \vee p_2 \vee \sim p_3$. Conjuncția disjuncțiilor pentru care formula este 0 va fi 0 și 1 pentru celelalte.

Conjuncțiile și disjuncțiile care au proprietatea de a lua valoarea formulei în cîte o interpertare a ei se numesc *minitermeni* sau *clauze conjunctive*, respectiv *maxitermeni* sau *clauze disjunctive*. Formarea lor este indicată de raționamentul utilizat. Minitermenii se formează scriind



în dreptul fiecărei interpretări o conjuncție, unde apare p_i cînd $(p_i)^v = 1$ și $\sim p_i$ cînd $(p_i)^v = 0$, iar disjuncțiile în care apare p_i cînd $(p_i)^v = 0$ și $\sim p_i$ cînd $(p_i)^v = 1$.

Vom scrie clauzele conjunctive și disjunctive ce corespund tablei de adevăr a unei formule cu trei variabile propoziționale :

p_1	p_2	p_3	P	Minitermeni	Maxitermeni
1	1	1	?	$p_1 \wedge p_2 \wedge p_3$	$\sim p_1 \vee \sim p_2 \vee \sim p_3$
1	1	0	?	$p_1 \wedge p_2 \wedge \sim p_3$	$\sim p_1 \vee \sim p_2 \vee \sim p_3$
1	0	1	?	$p_1 \wedge \sim p_2 \wedge p_3$	$\sim p_1 \vee p_2 \vee \sim p_3$
1	0	0	?	$p_1 \wedge \sim p_2 \wedge \sim p_3$	$\sim p_1 \vee p_2 \vee p_3$
0	1	1	?	$\sim p_1 \wedge p_2 \wedge p_3$	$p_1 \vee \sim p_2 \vee \sim p_3$
0	1	0	?	$\sim p_1 \wedge p_2 \wedge \sim p_3$	$p_1 \vee \sim p_2 \vee p_3$
0	0	1	?	$\sim p_1 \wedge \sim p_2 \wedge p_3$	$p_1 \vee p_2 \vee \sim p_3$
0	0	0	?	$\sim p_1 \wedge \sim p_2 \wedge \sim p_3$	$p_1 \vee p_2 \vee p_3$

Formula ce corespunde acestei table de adevăr este disjuncția tuturor minitermenilor ce corespund lui 1 în coloana P sau conjuncția tuturor maxitermenilor ce corespund lui 0 în coloana P.

Exemplu. Fie tabla de adevăr

p	q	r	
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0
0	0	0	1

Care este formula restrînsă care îi corespunde?

Cele trei evaluări pentru care formula căutată este adevărată sînt 1 1 1, 1 1 0 și 0 0 0. Cele trei conjuncții corespunzătoare sînt :

$$p \wedge q \wedge r, \qquad p \wedge q \wedge \sim r, \qquad \sim p \wedge \sim q \wedge \sim r,$$

iar rezultatul este disjuncția acestora :

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \sim r) \vee (\sim p \wedge \sim q \wedge \sim r).$$

La fel, întrucît maxitermenii ce corespund evaluărilor pentru care formula este 0 sînt $\sim p \vee q \vee \sim r$, $\sim p \vee q \vee r$, $p \vee \sim q \vee \sim r$, $p \vee \sim q \vee r$, $p \vee q \vee \sim r$, și deci formula se poate scrie

$$(\sim p \vee q \vee \sim r) \wedge (\sim p \vee q \vee \sim r) \wedge (p \vee \sim q \vee \sim r) \wedge (p \vee \sim q \vee r) \wedge (p \vee q \vee \sim r).$$

Rezultă următoarele două

Propoziții. 1. *Orice formulă care nu este o contradicție este logic echivalentă cu o formulă restrînsă normală disjunctivă :* $\bigvee_{i \leq m} \bigwedge_{j \leq n} l_{ij}$, *unde* l_{ij} *sînt literale.*

2. *Orice formulă care nu este o tautologie este logic echivalentă cu o formulă de formă normală conjunctivă :* $\bigwedge_{i \leq m} \bigvee_{j \leq n} l_{ij}$, *unde* l_{ij} *sînt literale.*

Scanned with OKEN Scanner

Demonstrație. Două formule sînt logic echivalente dacă și numai dacă corespund la aceeași funcție de adevăr.

În cazul unei formule date îi construim tabla de adevăr și procedăm ca mai sus, scriind forma disjunctiv normală.

Pentru propoziția 2, dacă formula dată P nu este o tautologie, atunci P nu este contradicție și este deci de forma $\sim(\bigvee_{i\leqslant m}\bigwedge_{j\leqslant n} l_{ij})$ conform propoziției 1. Aplicînd legile lui De Morgan, aceasta se poate scrie $\bigwedge_{i\leqslant m}\bigvee_{j\leqslant n}(\sim l_{ij})$ și dacă în această expresie aplicăm legea dublei negații, unde literalii sînt atomi negativi, obținem forma conjunctivă normală.

Prin intermediul tablei de adevăr putem scrie forma normală conjunctivă sau forma normală disjunctivă a unei formule. *Care este procedeul de a ajunge la una din formele normale a unei formule date fără să recurgem la tabla de adevăr?* Procedura este următoarea :

P a s u l 1. Elimină implicația (dublă sau simplă).

Se aplică formulele :

$$P \leftrightarrow Q \Leftrightarrow (P \to Q)\wedge(Q \to P),\quad P \to Q \Leftrightarrow \sim P \vee Q.$$

P a s u l 2. Mută negația.

Se aplică formulele dublei negații

$$\sim(\sim P) \Leftrightarrow P$$

și De Morgan

$$\sim(P \vee Q) \Leftrightarrow \sim P \wedge \sim Q,\quad \sim(P \wedge Q) \Leftrightarrow \sim P \vee \sim Q.$$

P a s u l 3. Aplică distributivitatea.

Se apelează formulele

$$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q)\wedge(P \vee R),$$
$$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q)\vee(P \wedge R).$$

Exemple. 1) Fie $P \Leftrightarrow ((p \leftrightarrow q) \to (r \to s))\wedge(q \to \sim(p \wedge r))$.

Prin eliminarea dublei implicații se obține

$$(((p \to q)\wedge(q \to p)) \to (r \to s))\wedge(q \to \sim(p \wedge r)).$$

Se elimină $\to$:

$$(\sim((\sim p \vee q)\wedge(\sim q \vee p))\vee(\sim r \vee s))\wedge(\sim q \vee \sim(p \wedge r)).$$

Se mută $\sim$:

$$(\sim(\sim p \vee q)\vee\sim(\sim q \vee p)\vee(\sim r \vee s))\wedge(\sim q \vee \sim p \vee \sim r)) \Leftrightarrow$$
$$\Leftrightarrow((p \wedge \sim q)\vee(q \wedge \sim p)\vee\sim r \vee s)\wedge(\sim q \vee \sim p \vee \sim r).$$

Prin aplicarea legii distributivității, obținem *FND* :

$$P \Leftrightarrow (p \wedge \sim q)\vee(p \wedge \sim q \wedge \sim r)\vee(q \wedge \sim p)\vee(q \wedge \sim p \wedge \sim r)\vee$$
$$\vee(\sim r \wedge \sim q)\vee(\sim r \wedge \sim p)\vee\sim r \vee(s \wedge \sim q)\vee(s \wedge \sim p)\vee(s \wedge \sim r).$$

2) Fie $Q \Leftrightarrow p \wedge (q \to r) \to s$. Avem

$$(p \wedge (\sim q \vee r)) \to s \Leftrightarrow \sim(p \wedge (\sim q \vee r))\vee s \Leftrightarrow$$
$$\Leftrightarrow \sim p \vee \sim(\sim q \vee r)\vee s \Leftrightarrow (\sim p \wedge (q \vee \sim r))\vee s \Leftrightarrow$$
$$\Leftrightarrow ((\sim p \vee q)\wedge(\sim p \vee \sim r))\vee s \Leftrightarrow$$
$$\Leftrightarrow (s \vee \sim p \vee q)\wedge(s \vee \sim p \vee \sim r),$$ care este FNC.

Scanned with OKEN Scanner

Formele normale, conjunctivă și disjunctivă, oferă a doua metodă eficientă pentru determinarea valorii de adevăr a unei formule, prima pe care am folosit-o fiind aceea a tablelor de adevăr.

Definiții. Fie l_i o secvență de literale distincte ($1 \leqslant i \leqslant n$). Formula $\bigvee_{i \leqslant n} l_i$ se numește o *clauză disjunctivă* sau clauză $\bigvee$, iar formula $\bigwedge_{i \leqslant n} l_i$ este numită *clauză conjunctivă* sau clauză $\bigwedge$. Pe viitor, termenul de clauză fără specificare va însemna clauza disjunctivă.

Vom utiliza termenul de *pereche opusă* pentru literalii l, l' legați de relația $l \Leftrightarrow \sim l'$.

Teoremă. *Fie l_i un literal ($1 \leqslant i \leqslant n$). Următoarele enunțuri sînt echivalente:*

1) $\bigvee_{i \leqslant n} l_i$ *este o tautologie;*

2) $\bigwedge_{i \leqslant n} l_i$ *este nerealizabilă;*

3) *există o pereche opusă* l_i, l_j ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant n$).

Demonstrație. Dacă $l_i = \sim l_j$, $\bigvee_{i \leqslant n} l_i$ este o tautologie, deoarece disjuncția conține pe $l_i \vee l_j = 1$ și $\bigwedge_{i \leqslant n} l_i$ este nerealizabilă, întrucît conjuncția conține pe $l_i \wedge l_j = 0$. Dacă nu există o pereche opusă de literali, există evaluări v, w, astfel că $v(l_j) = 1$, $w(l_j) = 0$ pentru $1 \leqslant i \leqslant n$. Atunci $(\bigvee_{i \leqslant n} l_i)^w = 0$ și $(\bigwedge_{i \leqslant n} l_i)^v = 1$, astfel că $\bigvee_{i \leqslant n} l_i$ nu este o tautologie, iar $\bigwedge_{i \leqslant n} l_i$ este realizabilă.

Am stabilit că există un algoritm care transformă orice formulă P într-o formulă P' în *FNC* astfel ca $P \Leftrightarrow P'$. Există un algoritm dual pentru *FND*.

Teoremă. *O formulă FNC este tautologie dacă și numai dacă fiecare din clauzele sale disjuncte este o tautologie. Dual, o formulă în FND este nerealizabilă dacă și numai dacă fiecare din clauzele sale conjunctive este nerealizabilă.*

Demonstrație. Fie $P = \bigwedge_{i \leqslant n} K_i$, unde K_i sînt clauze disjunctive. Dacă fiecare K_i este o tautologie pentru orice evaluare v, atunci $K_i^v = 1$ ($1 \leqslant i \leqslant n$) și $P^v = 1$. Dacă un K_i nu este o tautologie, există o evaluare v pentru care $K_i^v = 0$ și deci $P^v = 0$, nefiind o tautologie.

Această teoremă ne permite să determinăm valoarea de adevăr a unei formule, cercetînd clauzele formelor ei normale.

Exemplu. Fie $P \underset{\sim}{\Delta}$ crește cursa înarmărilor, $Q \underset{\sim}{\Delta}$ sentimentul de securitate scade, $R \underset{\sim}{\Delta}$ protestul pacifiștilor se intensifică. Să se arate că din $P \to Q$, $Q \to R$, P rezultă R.

Vom proceda prin tabla de adevăr a formulei

$$(P \to Q) \wedge (Q \to R) \wedge P \wedge \sim R.$$

Dacă R este concluzia logică a celor trei premise, această formulă este nerealizabilă.

P	Q	R	$P \to Q$	$Q \to R$	$\sim R$	$(P \to Q) \wedge (Q \to R) \wedge P \wedge \sim R$
1	1	1	1	1	0	0
1	1	0	1	0	1	0
1	0	1	0	1	0	0
1	0	0	0	1	1	0
0	1	1	1	1	0	0
0	1	0	1	0	1	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0

Scanned with OKEN Scanner

Vom recurge la FND a premiselor :

$$(P \to Q) \wedge (Q \to R) \wedge P \Leftrightarrow (\sim P \vee Q) \wedge (\sim Q \vee R) \wedge P \Leftrightarrow$$

$$\Leftrightarrow ((P \wedge \sim P) \vee (P \wedge Q)) \wedge (\sim Q \vee R) \Leftrightarrow (0 \vee (P \wedge Q)) \wedge (\sim Q \vee R) \Leftrightarrow$$

$$\Leftrightarrow (P \wedge Q \wedge \sim Q) \vee (P \wedge Q \wedge R) \Leftrightarrow (P \wedge 0) \vee (P \wedge Q \wedge R) \Leftrightarrow P \wedge Q \wedge R,$$

care este o clauză conjunctivă.

Premisele fiind adevărate, R este adevărat, fiind astfel o concluzie logică a acestora.

1.14. NOTĂ ISTORICĂ

Într-o prezentare a încercărilor de a automatiza raționamentul uman, astăzi calculul propozițional revendică primul loc prin faptul că este prima reușită de formalizare, din care s-au inspirat apoi logica predicatelor și logicile neclasice. În calculul propozițiilor, propozițiile nu se desfac, iar despre atomi nu știm decît că sînt adevărați sau falși. Nici o ipoteză nu se face asupra structurii propoziției simple, asupra subiectului și predicatului. Istoricește însă studiul judecăților cu predicație a fost primul abordat. Judecățile sînt împărțite încă de Aristotel în universale (afirmative și negative) și particulare (afirmative și negative) :

A : Toți S sînt P. E : Nici un S nu este P.

I : Unii S sînt P. O : Unii S nu sînt P.

Teoria silogismului constituie o tentativă de automatizare a raționamentului. Silogismele au fost clasate după figuri (funcția termenului mediu în premise) și apoi după moduri (după cantitatea și calitatea judecăților). Astfel, silogismele din figura întîi :

$$\begin{array}{ll} \text{Premisă majoră} & M - P \\ \text{Premisă minoră} & S - M \\ \hline \text{Concluzie} & S - P \end{array}$$

sînt

	Premisă majoră	Premisă minoră	Concluzie
BArbArA	*A*	*A*	*A*
CElArEnt	*E*	*A*	*E*
DArII	*A*	*I*	*I*
FErIO	*E*	*I*	*O*

Enunțul lui Aristotel „dacă A este enunțat de toți B și B despre toți C, atunci A trebuie enunțat despre toți C" este o lege logică formală care susține că, indiferent de conținutul propozițiilor (de interpretarea lor), concluzia silogismului „toți B sînt A" și „toți C sînt B" nu poate fi decît de forma „toți C sînt A". Procedeul mnemotehnic de a reține succesiunea $A - A - A$ prin cuvîntul „*BArbArA*" constituie o regulă mecanică, un element de algoritm. Regulile de deducție silogistică au avut o mare circulație în Evul Mediu.

Calculul propozițiilor simbolic, așa cum îl cunoaștem astăzi, are originea în lucrările lui George Boole : „Analiza matematică a logicii" (1847)

Scanned with OKEN Scanner

și „O cercetare a legilor gîndirii" (1854). Este suficient să spunem că tabla de adevăr a fost folosită pentru prima oară de C. S. Peirce (1839—1914), dar unii o atribuie lui Ernest Schröder (1841—1902). Aceasta nu înseamnă că multe din tautologiile calculului propozițiilor nu erau cunoscute anterior. Astfel legile ce poartă numele lui August de Morgan (1806—1878), întrucît acesta le-a stabilit în cadrul unei teorii a relațiilor, se găsesc în tratatele medievale (Strodus, sec. XIV), enunțate astfel : „Opposita contradictoria copulativae est una disjunctiva composita ex contradictoriis partium copulativae" ($\sim(x \wedge y) \Leftrightarrow \sim x \vee \sim y$).

Gottfried Wilhelm Leibniz (1646—1716) este privit drept părintele raționamentului automat pentru că a lansat ideea unui „calculus ratiocinator", care ar putea simula raționamentul, și a unei „characteristica universalis", adică a unui sistem de nomenclatură universală ce ar putea reda cunoștințele fundamentale nu prin cuvine, ci prin semne. Este interesant de observat că, în notații diferite de cea de astăzi, Leibniz cunoștea formule logice valide importante, printre care formulele

$$a \Rightarrow b \Leftrightarrow (a \Leftrightarrow a \wedge b) \text{ și } b \Rightarrow a \Leftrightarrow (a \vee b \Leftrightarrow b).$$

Cea mai „complicată" lege dedusă de el este

$$((a \to b) \wedge (c \to d)) \to ((a \vee c) \to (b \vee d)).$$

În ceea ce privește conectorii, implicația a fost discutată în antichitate în cadrul studiului raționamentului ipotetic. Philon din școala megarică a micilor socratici definește implicația drept implicație materială : adevărată în afară de cazul că prima propoziție este adevărată și a doua falsă. La Diodor, tot din școala megarică, implicația este definită în funcție de timp, pentru momentul t, ceea ce trimite la logica temporală de astăzi. Definiția implicației materiale este reluată de stoici (sec. III î.e.n.), dar Chrysippos adoptă implicația strictă, reluată de C.I. Lewis în secolul nostru pentru logica modală :

$p \to q$, dacă $\sim q$ este incompatibil cu p, $(p \wedge \sim q)^r = 0$.

Tot stoicilor li se datorează $p \to q = \sim p \vee q$.

După unii istorici ai logicii descoperirea faptului că conectorii pot fi definiți unii prin alții trebuie datată nu în Evul Mediu sau la Leibniz, ci în jurul anului 250 î.e.n.

1.15. EXERCIȚII ȘI PROBLEME

1. Să se scrie tablele de adevăr ale următoarelor formule :

a) $p \to (q \to r)$;

b) $(p \vee \sim q) \wedge r$;

c) $(p \leftrightarrow \sim q) \vee q$;

d) $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$;

e) $(p \vee q) \to \sim (\sim p \wedge (r \vee s))$;

f) $\sim((\sim p \wedge \sim q) \wedge (p \vee r))$.

2. Care din următoarele formule sînt tautologii ?

a) $p \to (q \to p)$;

b) $(p \wedge \sim q) \vee (q \wedge \sim r) \vee (r \wedge \sim p)$;

d) $((p \to q) \wedge r) \to ((p \to r) \wedge q)$;

e) $\sim(\sim(q \to \sim p) \to (p \to r)) \to \sim(p \wedge q)$.

3. Să se arate prin metoda lui Quine și metoda reducerii că următoarele formule sînt tautologii :

a) $\sim p \to (p \to q)$;

b) $\sim (p \to p) \to p$;

Scanned with OKEN Scanner

c) $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$;

d) $((p \to q) \wedge (q \to r)) \to (p \to r)$;

e) $((p \to q) \to p) \to p$.

4. Să se demonstreze următoarele echivalențe:

a) $((\sim p \wedge \sim q) \to (\sim r)) \Leftrightarrow (r \to (q \vee p))$;

b) $((\sim p \vee q) \to r) \Leftrightarrow ((p \wedge \sim q) \vee r)$;

c) $((\sim(p \vee \sim q)) \to (q \to r)) \Leftrightarrow \sim((\sim q \vee r) \to (q \to p)) \Leftrightarrow (q \to (p \vee r))$;

d) $((p \to q) \to \sim(q \to p)) \Leftrightarrow ((p \wedge \sim q) \vee (q \vee \sim p))$;

e) $((p \vee q) \vee r) \Leftrightarrow \sim(\sim p \wedge \sim q \wedge \sim r)$.

5. Să se aducă la formă normală conjunctivă formulele:

a) $p \leftrightarrow q$;

b) $((p \to q) \to r) \to s$;

c) $(p \to q) \to ((p \wedge r) \to (q \wedge r))$;

d) $(p \leftrightarrow q) \to ((p \to r) \to (q \to r))$;

e) $(p \leftrightarrow q) \to ((p \to r) \to (q \to r))$.

6. Să se găsească forme normale disjunctive echivalente cu formulele:

a) $q \to (p \to (q \to p))$;

b) $p \to (\sim q \vee r)$;

c) $(p \wedge q) \vee (\sim q \leftrightarrow r)$;

d) $s \to (s \wedge (p \to (\sim p \to q)))$;

e) $(p \to (p \leftrightarrow r)) \to (\sim(p \to q) \leftrightarrow \sim(p \to r))$.

7. Care din următoarele formule sînt valide?

a) $p \leftrightarrow p \vee p$;

b) $\sim p \to p \wedge q$;

c) $(p \to q) \leftrightarrow (\sim p \vee q)$;

d) $((p \leftrightarrow q) \leftrightarrow r) \leftrightarrow (p \leftrightarrow (q \leftrightarrow r))$;

e) $((\sim p \vee q) \to r) \leftrightarrow p \to q$.

8. Să se demonstreze că dacă o formulă conține n variabile, atunci are $n - 1$ conectori binari.

9. Să se demonstreze că perechile $\{\wedge, \vee\}$ nu sînt o mulțime adecvată de conectori. La fel pentru $\{\sim, \leftrightarrow\}$.

10. Care este validitatea următoarei forme de inferențe:

Dacă p este adevărat, atunci dacă q, atunci r.
Dacă r și s, atunci t.
Dacă u, atunci s este adevărat și t este fals.

Dacă ambii p și q sînt adevărați, u este fals.

11. O formulă care conține numai conectorul $\leftrightarrow$ ese tautologie dacă și numai dacă fiecare variabilă propozițională apare de un număr par de ori.

12 Dacă $P_1, P_2, \ldots, P_n \Rightarrow P$, atunci $P_1, \ldots, P_{n-1} \Rightarrow (P_n \to P)$.

BIBLIOGRAFIE

1. Bieltz, P. *Principiul dualității în logica formală*. București, Editura Științifică, 1974.
2. Enescu, G. *Logica simbolică*. București, Editura Științifică, 1971.
3. Hughes, G. E., Londey, D. G. *The Elements of Formal Logic*. London, Methuen, 1965.
4. Kemeny, J. K., Snell, J. L., Thompson, G. *Algèbre moderne et activités humaines*. Paris, Dunod, 1965.

Scanned with OKEN Scanner

5. Moisil, Gr. C. *Elemente de logică matematică și teoria mulțimilor*. București, Editura Științifică, 1968.
6. Novikov, P.S. *Elemente de logică matematică*. București, Etitura Științifică, 1966.
7. Suppes, P. *Introduction to Logic*. New York, Van Nostrand, 1957.

NOTE ISTORICE

1. Dimitriu, A. *Istoria logicii*. București, Editura Didactică și Pedagogică, 1969.
2. Kotarbinski, J. *Leçons sur l'histoire de la logique*, Warszawa, PWN, 1965.
3. Kneale, W., Kneale, M. *Dezvoltarea logicii*, vol. I, Cluj-Napoca, Dacia, 1974, vol. II, 1975.

Scanned with OKEN Scanner

2.

SISTEMUL DEDUCTIV AL CALCULULUI PROPOZIȚIILOR

2.1. DEMONSTRAȚIE ȘI DEDUCȚIE

Calculul propozițiilor clasic, pe scurt CP, începe prin a reprezenta propozițiile vorbirii curente prin simboluri atomice, a le atribui două valori de adevăr (1 și 0) și a forma cu ajutorul unor conectori formule, a căror valoare de adevăr urmează să fie dedusă din valoarea de adevăr a atomilor care compun formula. O asemenea abordare, care are drept concept central „adevărul logic" și care răspunde la întrebarea „este o formulă dată tautologie (adică adevărată în toate interpretările) sau nu?" este o abordare *semantică*. Spre deosebire de ea, abordarea *sintactică* are drept concept cheie „demonstrația logică" și răspunde la întrebarea „este o formulă dată o teoremă (adică demonstrabilă în cadrul sistemului) sau nu?".

Abordarea sintactică este susceptibilă de un grad înalt de formalizare. În capitolul precedent formalizarea a constat în faptul că decizia dacă o formulă este adevărată sau nu, depinde exclusiv de forma ei, orice referire la interpretarea simbolurilor fiind exclusă. Știm însă că p, q, r, ... sînt simboluri care țin locul unor propoziții sau enunțuri. În calculul sintactic pornim la drum fără această ipoteză. Simbolurile sînt inițial neinterpretate, servesc la construirea unui sistem deductiv abstract, îi dezvăluie structura și numai apoi, prin interpretarea ca propoziții, se obțin consecințe noi despre logica acestora.

Un *sistem formal* este definit de:

1) un alfabet de simboluri,
2) o mulțime de formule bine formate;
3) o mulțime de axiome;
4) reguli de deducție.

Sistemul formal L este definit astfel:

1) **alfabetul** simbolurilor este infinit, $\sim$, $\rightarrow$, (,), p_1, p_2, ...;
2) **o formulă bine formată** (pe scurt fbf) este definită recursiv:
   a) p_i este o formulă fbf;
   b) dacă A și B sînt formule fbf, $(\sim A)$ și $(A \rightarrow B)$ sînt formule fbf;
   c) mulțimea formulelor fbf este generată numai prin regulile a) și b);
3) oricare ar fi formulele fbf A, B, C, următoarele formule fbf sînt **axiomele** lui L:

L1 $(A \rightarrow (B \rightarrow A))$,

L2 $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$,

L3 $(((\sim A) \rightarrow (\sim B)) \rightarrow (B \rightarrow A))$.

Scanned with OKEN Scanner

4) **Regula de inferență** este MP (modus ponens) : oricare ar fi formulele fbf A, B, din A și $(A \to B)$ rezultă B.

Prima noțiune care este introdusă în cadrul unui sistem formal este aceea de *demonstrație*. În cele ce urmează vom renunța la precizarea fbf înțelegînd că termenul „formulă" este definit de 2).

Definiție. Se numește *demonstrație* secvența de formule $A_1, A_2, \ldots, A_n$ astfel că :

1) A_i $(1 \leqslant i \leqslant n)$ este axiomă sau

2) A_i urmează din A_j, A_k $(j < i, k < i)$ două formule anterioare ale secvenței prin aplicarea lui MP.

Ultima formulă A_n este o teoremă în L, iar secvența este demonstrația lui A_n în L.

Exemplu. Fie $A_1 = p_1 \to (p_2 \to p_1)$,

$$A_2 = (p_1 \to (p_2 \to p_1)) \to ((p_1 \to p_2) \to (p_1 \to p_1)),$$

$$A_3 = (p_1 \to p_2) \to (p_1 \to p_1).$$

Deci, A_3 este o teoremă în L, iar A_1, A_2, A_3 demonstrația lui A_3 în L.

Într-adevăr, întrucît axiomele sînt scrise pentru orice A, B, C din L, este suficient să punem $A = p_1$, $B = p_2$, $C = p_1$ și se obțin :

L1 $\quad p_1 \to (p_2 \to p_1)$,

L2 $\quad (p_1 \to (p_2 \to p_1)) \to ((p_1 \to p_2) \to (p_1 \to p_1))$.

Aplicăm acum regula MP formulelor $A = p_1 \to (p_2 \to p_1)$ și $B = (p_1 \to p_2) \to (p_1 \to p_1)$. Din A afirmat de L1 și din $A \to B$ afirmat de L2, deducem B, adică

$$(p_1 \to p_2) \to (p_1 \to p_1), \qquad \text{MP } (L1, L2)$$

Noțiunea de demonstrație este lărgită la cea de deducție.

Definiție. Fie o mulțime de formule $\Gamma \in L$. Se numește *deducție din* Γ o secvență $A_1, \ldots, A_m$, astfel că :

1) A_i este o axiomă $(1 \leqslant i \leqslant n)$ sau

2) A_i este un membru al lui Γ sau

3) A_i urmează prin MP din doi membri antecedenți ai șirului A_j, A_k $(j < i, k < i)$.

Spunem că ultimul termen A_n este deductibil din Γ sau consecință a lui Γ și scriem

$$\Gamma \underset{L}{\vdash} A_n.$$

Reamintim că, din punct de vedere semantic, A_n este o consecință logică a unei mulțimi Γ de formule $A_1, A_2, \ldots, A_{n-1}$ cînd $A_1 \wedge A_2 \wedge \ldots \wedge A_{n-1} \to A_n$ este o tautologie, fapt care se notează $\Gamma \models A_n$.

Orice demonstrație este o deducție în care Γ este o mulțime vidă și deducția se face numai din axiome. O asemenea demonstrație produce o teoremă, deci în cazul unei teoreme A putem scrie $\emptyset \underset{L}{\vdash} A$ sau pe scurt $\underset{L}{\vdash} A$. Reamintim că proprietatea unei formule de a fi tautologie se notează $\models A$.

Scanned with OKEN Scanner

Exemple. 1) Fie $A_1 = A \to (B \to A)$,

$$A_2 = (A \to (B \to C)) \to ((A \to B) \to (A \to C)),$$

$A_3 = A$, $A_4 = B \to (A \to C)$ și $A_5 = B \to C$.

Secvența A_1 A_2, A_3, A_4, A_5 este o deducție în L. Este suficient să comparăm pe A_i cu tabloul de axiome și să vedem că A_1 și A_2 sînt axiomele L1 și L2, Γ este formată din A_3 și A_4, iar consecința lui Γ este A_5. Omițînd axiomele, vom scrie propoziția pe care vrem s-o demonstrăm:

$$(A, (B \to (A \to C))) \underset{L}{\vdash} B \to C.$$

Deducția are forma următoare:

(1) A

(2) $B \to (A \to C)$

(3) $A \to (B \to A)$ (L1)

(4) $B \to A$ MP (1, 3)

(5) $(B \to (A \to C)) \to ((B \to A) \to (B \to C))$ (L2)

(6) $(B \to A) \to (B \to C)$ MP (2, 5)

(7) $B \to C$ MP (4, 6)

Concluzia $B \to C$ este obținută prin aplicarea exclusivă a regulii MP asupra axiomelor și a altor formule; este o formulă în L, dar nu este o teoremă, după cum nu sînt teoreme formulele A, B, C, ...

2) Să se demonstreze teorema în L:

$$\underset{L}{\vdash} (A \to A).$$

Teorema trebuie obținută numai din axiome. Vor fi suficiente L1 și L2, scrise pentru anumite A, B, C, convenabil alese:

(1) $A \to ((A \to A) \to A)$, L1, unde B este $A \to A$,

(2) $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$,

L2, unde B este $A \to A$ și C este A,

(3) $(A \to (A \to A)) \to (A \to A)$, MP (1, 2),

(4) $A \to (A \to A)$, L1, unde B este A,

(5) $A \to A$, MP (3, 4).

3) Este $\sim B \to (B \to A)$ o teoremă în L?

Formăm demonstrația utilizînd toate trei axiome ale sistemului L:

(1) $\sim B \to (\sim A \to \sim B)$, L1, unde A este $\sim B$, B este $\sim A$.

(2) $(\sim A \to \sim B) \to (B \to A)$, L3,

(3) $((\sim A \to \sim B) \to (B \to A)) \to (\sim B \to ((\sim A \to \sim B) \to (B \to A)))$,

L1, unde A este $(\sim A \to \sim B) \to (B \to A)$, iar B este $\sim B$,

(4) $\sim B \to ((\sim A \to \sim B) \to (B \to A))$, MP (2, 3),

(5) $(\sim B \to ((\sim A \to \sim B) \to (B \to A))) \to ((\sim B \to (\sim A \to \sim B)) \to (\sim B \to$

$\to (B \to A)))$, L2, unde A este $\sim B$, B este $\sim A \to \sim B$, C este $B \to A$,

(6) $(\sim B \to (\sim A \to \sim B)) \to (\sim B \to (B \to A))$, MP (4, 5),

(7) $\sim B \to (B \to A)$, MP (1, 6).

Sîntem astfel îndreptățiți să scriem: $\underset{L}{\vdash} (\sim B \to (B \to A))$.

Scanned with OKEN Scanner

2.2. COMENTARII ASUPRA CONSTRUIRII SISTEMULUI *L*

Alfabetul. Preocuparea de economicitate a sistemului formal se recunoaște din acceptarea unui număr minimal de simboluri, necesare construirii sistemului. În cap. 1 am folosit un număr mare de simboluri: din 16 conectori binari posibili am ales patru ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), la care am adăugat operatorul monadic $\sim$. O mulțime de conectori adecvați este însă suficientă, operațiile celorlalți putînd fi definite prin primii. Perechea ($\sim$, $\rightarrow$) este o pereche adecvată, deci este suficient să o reținem pe ea în alfabet.

În vocabularul inițial nu figurează simbolurile $\vdash$ și $\models$. Înseamnă că orice formulă care conține unul din aceste două semne nu este o formulă aparținînd sistemului *L*, ci este o propoziție despre formulele sistemului. Simbolurile, enunțurile, teoremele, care nu sînt în sistem, dar se utilizează la descrierea lui, sînt desemnate de prefixul **meta** (de la grecescul „după" sau „în afară"). Mai general, logica ce ne permite să analizăm logica este o *metalogică*, iar matematica ce se ocupă de structurile matematicii este o *metamatematică*.

Orice formulă trebuie să fie scrisă în paranteză. Astfel, riguros, numai $(\sim A)$ este o formulă, în timp ce $\sim A$ nu este. Dar utilizarea strictă a parantezei duce la un număr de paranteze greu de mînuit în scrierea de formule; de aceea le omitem ori de cîte ori operațiile ce urmează să fie efectuate de conectori în formulă și succesiunea lor sînt clare.

Observăm că simbolurile A, B, C, ... nu figurează în alfabetul sistemului *L*. Ele nu fac deci parte din limbajul acestui sistem și sînt utilizate pentru a putea vorbi **despre** el. Acestea desemnează formule ale sistemului, scrise cu simbolurile lui, adică A poate fi înlocuit cu oricare din formulele $p_1 \rightarrow p_2$, $(p_1 \rightarrow p_2) \rightarrow (\sim p_3 \rightarrow p_4)$ etc. De aceea ele se numesc și *variabile metalogice*, avînd ca domeniu formulele sistemului. Limbajul care rezultă din simbolurile sistemului este un *limbaj obiect*, în timp ce limbajul construit cu *metasimboluri* este un *metalimbaj*.

Construirea sistemului *L* se face cu intenția ca atunci cînd îl vom interpreta adecvat (vom da un sens simbolurilor sale), vom găsi calculul propozițiilor. De aceea alegerea simbolurilor este o selecție riguroasă și minimală, aplicată alfabetului folosit și testat în logica simbolică. Nici un simbol inclus în alfabet nu are vreo proprietate, în afara celor conținute în definiția lui *L* sau derivate din ea. Astfel conectorii $\sim$ și $\rightarrow$ sînt definiți exclusiv prin efectele lor în formarea de formule bine formate (punctul 2 al definirii sistemului formal), prin axiome (punctul 3) și prin efectele lor în deducerea de formule fbf noi. Dacă includem în alfabetul de simboluri și un alt operator (de pildă, disjuncția), trebuie să precizăm rolul său în formarea de formule fbf, să avem axiome care să arate aplicarea lui corectă și legătura cu alți conectori, să precizăm efectele sale atunci cînd îl utilizăm în reguli de deducție.

Conectorii conjuncției și disjuncției și ai dublei implicații pot fi folosiți în scrierea formulelor într-un sistem formal dacă sînt introduși prin definiții, dar demonstrațiile se fac cu simbolurile care figurează în axiome și în regulile de inferență.

Formulele bine formate. Denumirea de „bine formate" pe care multe cărți de logică le aplică consecvent, nu este destinată decît pentru a preciza că formulele se conformează regulilor din definiție. O secvență finită de

simboluri este o *expresie*. Mulțimea formulelor F constituite conform regulilor este o submulțime proprie a mulțimii expresiilor E. Formula $((p_1 \to p_2) \to (p_1 \to p_3))$ este fbf și aparține lui L. Ea este o instanțiere a formulei $(A \to B)$, care este o formulă bine formată, dar este „despre" formulei sistemului, folosind variabilele metalogice; o numim *schemă* pentru a o distinge de o formulă propriu-zisă.

Numai instanțierile variabilelor metalogice în formulele fbf metalogice pot aparține lui L.

Fie $\{A, (B \to (A \to C))\} \vdash_L (B \to C)$. Aceasta nu este o formulă fbf deoarece conține simboluri care nu aparțin alfabetului, ca $\{,\}$ și $\vdash$. Ea este totuși o formulă a metalimbajului, prin care descriem gramatica sau sintaxa sistemului L.

Întrucît interesul este îndreptat spre posibilitatea de a stabili care formule fbf sînt teoreme în sistemul formal, distincția anterioară între formulele fbf ale sistemului și formulele limbajului despre sistem se face și mai nuanțat în cadrul teoremelor. Vom ilustra prin exemplul deducției.

Fie $\{A, (B \to (A \to C))\} \vdash_L (B \to C)$. Acesta este un enunț care nu constituie o formulă fbf, dar care spune ceva despre sistemul L și anume că $B \to C$ este dedus din alte două formule fbf. Acest rezultat nu este o teoremă, fiindcă nu este obținut prin demonstrație din axiome. Dar enunțul în metalimbaj asupra deducției de mai înainte este totuși un enunț valabil pentru formulele fbf din L. El se numește o *metateoremă* și constituie un rezultat despre sistemul L. Asemenea enunțuri au fost formalizate în cadrul calculului cu reguli de inferență (v. 2.3).

În prezenta carte, majoritatea teoremelor stabilite în calculul propozițiilor sînt metateoreme. Mulți autori încearcă să evite confuzia, numind metateoremele „propoziții", dar ei continuă să le „demonstreze", deși demonstrarea metateoremelor nu coincide cu demonstrația definită în 2.1, ci este în fond deducție. Important este că la fiecare „teoremă" să putem spune dacă este o teoremă a sistemului L sau o metateoremă despre L.

Axiomele. Cele trei axiome L1, L2, L3 ale sitemului formal L sînt, în realitate, o infinitate de axiome, pentru că sînt enunțate în variabile metalogice și pot fi instanțiate într-o infinitate de moduri. De aceea ele se numesc, mai corect, *scheme de axiome*. O dată cu schema L1 $A \to (B \to A)$, avem axiomele

$$p_1 \to (p_2 \to p_1),\ p_2 \to (p_3 \to p_2),\ (p_1 \to p_2) \to (p_3 \to (p_1 \to p_2)) \quad \text{etc.},$$

adică toate axiomele obținute din $A \to (B \to A)$ punînd în locul lui A, B toate formulele fbf ale sistemului L.

Putem proceda și altfel: se enunță o axiomă în limbajul sistemului, de exemplu L1 $p_1 \to (p_2 \to p_1)$, și se instituie o *regulă de substituție* care se include în definiția lui L și permite ca prin substituția simbolurilor cu orice formulă fbf, cu condiția înlocuirii fiecărui simbol cu aceeași formulă peste tot unde apare, să obținem o formulă fbf cu același caracter ca formula fbf inițială, în cazul de față, tot o axiomă.

Axiomele sînt formulele fbf care generează toate teoremele. Dar care este geneza lor? Sistemul formal al calculului propozițiilor este precedat de semantica sa. Acolo, în prezența unui număr infinit de tautologii, se naște întrebarea: care tautologii pot fi selecționate ca un nucleu

Scanned with OKEN Scanner

minimal, astfel ca în cazul introducerii unei reguli de deducție, care păstrează caracterul de adevăr, să obținem celelalte tautologii? Pentru ca să existe posibilitatea racordării sistemului formal la calculul semantic al propozițiilor, axiomele trebuie să se numere printre tautologiile acestuia. Verificarea este ușoară în cazul axiomelor L1, L2, L3.

Preocuparea economicității sistemului la stabilirea alfabetului este prezentă și la alegerea axiomelor. Ele trebuie să satisfacă criteriul numărului minimal de axiome.

Regulile de deducție (inferență) Regulile de deducție adoptate de cele mai multe sisteme formale sînt substituția și MP. Ambele au fost definite în semantica calculului propozițiilor, scoțindu-se la iveală o proprietate comună : ele păstrează caracterul de tautologie pentru propoziția obținută prin aplicarea lor. Aceste două reguli nu erau singurele posibile cu această proprietate. Astfel prin *adjuncție* (legarea prin conjuncție a două formule care sînt tautologii) se obține tot o tautologie. Numim *compoziție* operația de obținere a formulei $A \to BC$ din $A \to B$ și $A \to C$. *Substituția prin echivalență* este obținerea unei formule noi dintr-o formulă dată prin substituția unei variabile propoziționale cu o formulă echivalentă. Deosebirile față de substituția uniformă folosită pînă aici sînt :

a) la substituția uniformă, o variabilă se înlocuiește peste tot cu orice formulă (echivalentă cu ea sau nu),

b) la substituția prin echivalență putem înlocui fiecare apariție a unei variabile cu cîte o altă formulă, cu condiția ca toate să fie echivalente cu ea.

Toate aceste reguli de obținere de noi propoziții se numesc *inferențe*. Este de reținut că regulile de deducție sînt *reguli de inferență*. Aproape toate sistemele au încorporat două reguli : substituția și MP (detașarea). Ele sînt scrise, ca și axiomele, cu ajutorul unor scheme, folosind variabilele metalogice $A1$, $A2$, ... în forma următoare :

1) Modus ponens
$$\frac{A \to B \qquad A}{B}$$

un alt mod de a scrie $\{A, A \to B\} \vdash B$.

2) Substituția
$$\frac{A(p_1, p_2, \ldots, p_n)}{A(B_1/p_1, B_2/p_2, \ldots, B_n/p_n)}$$

Cu alte cuvinte, dacă $A(p_1, p_2, \ldots, p_n)$ este deductibilă din sistem, atunci și $A(B_1/p_1, \ldots, B_n/p_n)$ este deductibilă, unde operația B_i/p_i înseamnă substituția variabilei formale p_i prin formula fbf, B_i.

2.3. SISTEMELE CALCULULUI PROPOZIȚIONAL CLASIC

Este edificator pentru construirea sistemului formal să enumerăm sistemele definite de alte mulțimi de axiome.

a) **Sistemul lui Frege** are drept axiome :

F1 $p \to (q \to p)$,

F2 $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$,

Scanned with OKEN Scanner

F3 $(p \to (q \to r)) \to (q \to (p \to r))$,

F4 $(p \to q) \to (\sim q \to \sim p)$,

F5 $\sim \sim p \to p$.

F6 $p \to \sim \sim p$.

Enunțate în simbolurile limbajului, axiomele sînt însoțite de regula substituției și de regula MP.

b) **Sistemul lui Lukasiewicz** este sistemul **L** considerat mai înainte. El s-a obținut din sistemul lui Frege, reținînd F1 și F2. Axioma L3 este o formă modificată a lui F4. Celelalte axiome F3, F5, F6 pot fi deduse din L1, L2, L3 și, nefiind deci independente, pot fi omise :

L1 $p \to (q \to p)$,

L2 $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$,

L3 $(\sim p \to \sim q) \to (q \to p)$.

c) **Sistemul lui Russell** a apărut în Principia Mathematica (1910) cu următoarele axiome, numite principii :

R1 $(p \vee p) \to p$, *principiul tautologiei*,

R2 $p \to (p \vee q)$, *principiul adițiunii*,

R3 $(p \vee q) \to (q \vee p)$, *principiul permutării*,

R4 $(p \vee (q \vee r)) \to (q \vee (p \vee r))$, *principiul asociației*,

R5 $(p \to q) \to ((r \vee p) \to (r \vee q))$, *principiul sumei*.

Alfabetul conține mai mulți conectori, dar perechea adecvată este considerată $\{\vee, \sim\}$. Implicația $p \to q$ apare în axiome ca notație pentru $\sim p \vee q$. Ceilalți conectori definiți prin $\sim$, $\vee$ sînt $\to$, $\wedge$, $\leftrightarrow$, $|$, $\downarrow$.

Regulile de deducție sînt MP și substituția.

d) **Sistemul lui Hilbert-Ackermann** are drept axiome axiomele lui Russell fără R4, omisă ca nefiind independentă. Rescriem :

H1 $(p \vee p) \to p$, H3 : $(p \vee q) \to (q \vee p)$,

H2 $p \to (p \vee q)$, H4 : $(p \to q) \to ((r \vee p) \to (r \vee q))$.

e) **Sistemul lui Church** este construit cu un conector $\to$ și cu constanta f (falsul). Sînt reținute primele două axiome ale lui Lukasiewicz și înlocuită a treia cu o axiomă ce leagă conectorul de f :

C1 $p \to (q \to p)$,

C2 $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$,

C3 $((p \to f) \to f) \to p$.

f) **Sistemul lui Nicod** nu reține în alfabetul său decît un conector $|$. Sistemul este construit cu o singură axiomă :

N : $(p|(q|r)) \mid ((s|(s|s)) \mid ((t|q)|((p|t)|(p|t))))$.

Desigur, în locul MP scris cu $\to$ este adoptată o altă regulă de deducție.

Scanned with OKEN Scanner

Observăm că ori de cîte ori o axiomă s-a dovedit a nu fi independentă, ea a fost abandonată. Preocuparea pentru formarea unei mulțimi cu cel mai redus număr de axiome a dus la elaborarea unei reguli practice de verificare rapidă a independenței axiomelor.

2.4. INDEPENDENȚA AXIOMELOR

Procedeul de verificare a independenței axiomelor constă în găsirea unei interpretări, care substituie atomilor valorile dintr-o *mulțime de interpretare* (de exemplu $M = \{1, 2, 3\}$), din construirea unei interpretări pentru formulele obținute prin conectori, alegerea unei valori de interpretare (de exemplu 1), și din determinarea valorilor care revin formulelor axiome în interpretarea aleasă. Dacă pentru o axiomă nu se obține valoarea desemnată, în timp ce celelalte axiome iau valoarea desemnată, înseamnă că axioma este independentă de celelalte, neavînd proprietatea de validitate a celorlalte. Vom ilustra metoda prin următorul

Exemplu. Fie sistemul Hilbert-Ackermann. În interpretarea

p	$\sim p$
1	3
2	2
3	1

și

		q		
	$\vee$	1	2	3
	1	1	1	1
p	2	1	1	2
	3	1	2	3

în mod necesar din definiția $p \to q = \sim p \vee q$ rezultă interpretarea pentru $p \to q$:

		q		
	$\to$	1	2	3
	1	1	2	3
p	2	1	1	2
	3	1	1	1

Aplicînd la axioma H2, obținem

		q		
	$p \to (p \vee q)$	1	2	3
	1	1	1	1
p	2	1	1	1
	3	1	1	1

La fel celelalte H3, H4 iau valoarea desemnată 1 în interpretarea de mai sus. Nu același lucru se întîmplă cu H1:

p	$(p \vee p) \to p$
1	1
2	2
3	1

Scanned with OKEN Scanner

Deci H1 este independentă de H2, H3, H4.
Interpretarea $M = \{1, 2, 3, 4\}$ și valoarea desemnată 1 :

p	$\sim p$
1	4
2	3
3	2
4	1

și

$p \vee q$	1	2	3	4
1	1	1	1	1
2	1	4	1	4
3	1	1	4	4
4	1	1	4	4

va demonstra independența axiomei H2.
Interpretarea $M = \{1, 2, 3, 4\}$, valoarea desemnată 1 și

p	$\sim p$
1	4
2	3
3	2
4	1

$p \vee q$	1	2	3	4
1	1	1	1	1
2	1	2	1	2
3	1	1	3	4
4	1	2	3	4

va demonstra independența axiomei H3.
Din interpretarea $M = \{1, 2, 3, 4\}$, valoarea desemnată 1 și

p	$\sim p$
1	2
2	1
3	4
4	1

$p \vee q$	1	2	3	4
1	1	1	1	1
2	1	2	3	4
3	1	3	3	1
4	1	4	1	4

va rezulta independența axiomei H4.

2.5. TEOREMA DEDUCȚIEI

Teorema deducției este în realitate o metateoremă, menită să faciliteze tehnica deducției, avînd rolul unei reguli suplimentare de inferență. Ea se enunță astfel :

Dacă $\Gamma \cup \{A\} \underset{L}{\vdash} B$, *atunci* $\Gamma \underset{L}{\vdash} (A \to B)$, *unde* Γ *este o mulțime, nu neapărat nevidă, de formule* fbf, *iar* A *și* B *sînt două formule.*

Demonstrație. Procedăm prin inducție, considerînd că numărul formulelor din secvența deducției lui B este 1. În acest caz, trebuie luate în considerare ipotezele :

a) B este axiomă în L sau este membru al lui Γ. În ambele cazuri formăm deducția

(1) B, axiomă sau $B \in \Gamma$,
(2) $B \to (A \to B)$, L1,
(3) $A \to B$, MP (1, 2),

deci putem scrie $\Gamma \underset{L}{\vdash} (A \to B)$.

Scanned with OKEN Scanner

b) B este A. Recurgem la demonstrația $\underset{L}{\vdash} A \to A$ și o folosim drept deducție $\Gamma \underset{L}{\vdash} (A \to A)$.

Pentru cazul în care secvența deducției are n membri ($n > 1$), presupunem că teorema este valabilă pentru toate formulele C deductibile din $\Gamma \cup \{A\}$ printr-un șir cu mai puțin de n membri.

a') Ipoteza că B este axiomă în L sau face parte din Γ ne conduce la o demonstrație identică cu cea de la punctul a).

b') Ipoteza că B este A se reduce la raționamentul b).

c') Ipoteza că B este obținută din două formule antecedente ale secvenței prin MP este tratată astfel : aceste două formule sînt în mod necesar de forma C și $C \to B$. Ambele sînt deduse, din cauza poziției lor în secvență, printr-o secvență cu mai puțin de n membri.

Putem afirma deci pentru acestea

$$\Gamma \cup \{A\} \underset{L}{\vdash} C \quad \text{și} \quad \Gamma \cup \{A\} \underset{L}{\vdash} C \to B$$

și, aplicînd teorema deducției valabilă pentru cazul $< n$,

$$\Gamma \underset{L}{\vdash} (A \to C) \quad \text{și} \quad \Gamma \underset{L}{\vdash} A \to (C \to B).$$

Dar secvența deducției care este obiectul demonstrației se termină în $A \to B$. Formăm deci această secvență alăturînd la secvența deducției din Γ a lui $A \to C$ secvența deducției din Γ a lui $A \to (C \to B)$ și continuînd astfel :

$(A \to (C \to B)) \to ((A \to C) \to (A \to B))$, L2,

$(A \to C) \to (A \to B)$, MP cu $A \to (C \to B)$,

$A \to B$, MP cu $A \to C$.

Am stabilit astfel, că de la orice ipoteză am pleca $\Gamma \underset{L}{\vdash} (A \to B)$.

Reciproca teoremei deducției. *Dacă* $\Gamma \underset{L}{\vdash} (A \to B)$, *atunci* $\Gamma \cup \{A\} \underset{L}{\vdash} B$, *oricare ar fi formulele* A, B *și mulțimea* (*posibil vidă*) Γ.

Demonstrație. Formăm secvența deducției lui $A \to B$ din Γ, adăugăm A ca membru al lui $\Gamma \cup \{A\}$ și aplicăm MP lui A și $A \to B$. Obținem B ca o consecință a lui $\Gamma \cup \{A\}$, deci $\Gamma \cup \{A\} \underset{L}{\vdash} B$.

Regula S1 (*silogismul ipotetic*) se obține din teorema deducției :

$$\{(A \to B), \quad (B \to C)\} \underset{L}{\vdash} (A \to C),$$

oricare ar fi A, B, C în L.

Formăm deducția :

(1) $A \to B$, ipoteză,

(2) $B \to C$, ipoteză,

(3) A, ipoteză,

(4) B, MP(1, 3),

(5) C, MP (2, 4).

Se obține concluzia $\{(A \to B), (B \to C), A\} \underset{L}{\vdash} C$, pe care o scriem sub forma

$$\{(A \to B), (B \to C)\} \cup \{A\} \underset{L}{\vdash} C$$

Scanned with OKEN Scanner

și, aplicînd teorema deducției, rezultă

$$\{(A \to B), (B \to C)\} \underset{L}{\vdash} (A \to C).$$

Noua regulă poate servi la demonstrarea unor teoreme, scurtînd calea demonstrației, în care ne-am fi servit doar de MP.

Exemplu. Fie $\underset{L}{\vdash} (\sim B \to (B \to A))$. Formăm demonstrația

(1) $\sim B \to (\sim A \to \sim B)$, L1,

(2) $(\sim A \to \sim B) \to (B \to A)$, L3,

(3) $\sim B \to (B \to A)$,

2.6. CONSISTENȚĂ, COMPLETITUDINE

Problema următoare este de a stabili dacă sistemul formal elaborat poate fi interpretat ca o logică a propozițiilor, cu alte cuvinte, de a face o legătură între sintaxa și semantica acestei logici. În acest scop este nevoie să introducem noțiunea de valoare de adevăr în sistemul formal, să definim cu ajutorul ei care formule sînt tautologii și să examinăm dacă tautologiile (semantice) coincid cu teoremele sintactice. Acest rezultat va fi enunțat astfel: dacă $\models p$, atunci $\vdash p$ și reciproc.

Definiții. *O evaluare* în sistemul formal L este o funcție care aplică orice formulă pe mulțimea $\{1, 0\}$ astfel ca

$v(A) \neq v(\sim A)$ și $v(A \to B) = 0$ dacă și numai dacă $v(A) = 1$ și $v(B) = 0$. Dacă pentru fiecare evaluare v avem $v(A) = 1$, atunci A este o tautologie. Primul rezultat este:

Orice teoremă a lui L este tautologie.

Demonstrație. Procedăm prin inducție asupra numărului de formule în secvența demonstrației teoremei A.

Dacă această secvență conține o singură formulă, A este o axiomă și deci tautologie.

Presupunem că orice teoremă a cărei secvența de demonstrație este de lungime $< n$ este tautologie. Fie A o teoremă a cărei secvență de demonstrație are n formule. Conform definiției, A este sau o axiomă (deci tautologie), sau provine din două formule prin aplicarea regulii MP, formule care trebuie să fie de forma B și $B \to A$.

Dar prin locul pe care îl ocupă în secvență ele sînt demonstrabile printr-o secvență de lungime $< n$, deci sînt tautologii. Dacă B și $B \to A$ sînt tautologii, A este tautologie. Dacă A nu ar fi tautologie, ar exista o evaluare pentru care $v(A) = 0$ și care împreună cu $v(B) = 1$ (este tautologie) ar duce la $v(B \to A) = 0$, contrar faptului că $v(B \to A) = 1$ ($B \to A$ este tautologie).

Pentru a demonstra reciproca acestei propoziții (orice tautologie este o teoremă în L) avem nevoie de următoarele concepte noi:

a) *Extensia L^** a unui sistem formal L este sistemul formal obținut prin modificarea sau lărgirea mulțimii de axiome a lui L, astfel că toate teoremele lui L să fie și teoreme ale lui L^*.

Scanned with OKEN Scanner

b) Extensia L^* este *consistentă* dacă nu există nici o formulă A astfel ca $\underset{L^*}{\vdash} A$ și $\underset{L^*}{\vdash} \sim A$ (A și $\sim A$ nu pot fi ambele teoreme în L^*).

c) Extensia L^* este *completă* dacă pentru fiecare A, avem $\underset{L^*}{\vdash} A$ sau $\underset{L^*}{\vdash} \sim A$ (cel puțin una din A sau $\sim A$ sînt teoreme în L^*).

Obținem următoarele noi proprietăți ale lui L și ale extensiunilor sale :

L este consistent.

Demonstrație. Dacă n-ar fi consistent, ar exista A astfel ca $\underset{L}{\vdash} A$ și $\underset{L}{\vdash} \sim A$. Dar A și $\sim A$, fiind teoreme, sînt tautologii. Dacă A este tautologie, $\sim A$ este o contradicție și invers. Deci A și $\sim A$ nu pot fi simultan teoreme și L este consistent.

L^* este consistent dacă și numai dacă există o formulă care nu este teoremă în L^*.

Demonstrație. Dacă L^* este consistent, oricare ar fi A, avem $\underset{L^*}{\vdash} A$ sau $\underset{L^*}{\vdash} \sim A$. Deci una din ele nu este teoremă în L^*.

Invers, presupunînd că L^* este inconsistent, putem arăta că nu există formulă, care să nu fie teoremă în L^*; cu alte cuvinte, fiecare formulă este teoremă în L^*. Fie L^* inconsistent. Atunci există B, astfel ca $\underset{L^*}{\vdash} B$ și $\underset{L^*}{\vdash} \sim B$. Fie A o formulă oarecare. Dar $\underset{L}{\vdash} (\sim B \to (B \to A))$ și L^* fiind extensia lui L : $\underset{L^*}{\vdash} (\sim B \to (B \to A))$. Scriem acum fragmentul de demonstrație :

(1) $\sim B$,

(2) $\sim B \to (B \to A)$,

(3) $B \to A$, MP (1, 2),

(4) B,

(5) A, MP (4, 3).

Deci oricare formulă A este teoremă într-un sistem inconsistent : $\underset{L^*}{\vdash} A$.

Extensia L^{**} obținută din extensia L^* consistentă a lui L, prin includerea lui $\sim A$ în L^{**} ca axiomă, A fiind o formulă a lui L care nu este teoremă a lui L^*, este consistentă.

Demonstrație. Dacă L^{**} nu ar fi consistent, ar exista o formulă $\underset{L^{**}}{\vdash} B$ și $\underset{L^{**}}{\vdash} \sim B$. Fie o formulă A oarecare. Atunci, ca și în demonstrația precedentă, A este teoremă. Scriem $\underset{L^{**}}{\vdash} A$. Întrucît diferența între L^* și L^{**} este că acesta din urmă are pe $\sim A$ drept axiomă suplimentară, o demonstrație în L^{**} este o deducție în L^*. Sîntem astfel îndreptățiți să scriem $\sim A \underset{L^*}{\vdash} A$ și prin teorema deducției $\underset{L^*}{\vdash} (\sim A) \to A$.

Dar $((\sim A) \to A) \to A$ este o teoremă în L, deci și în extesian sa L^* : $\underset{L^*}{\vdash} (((\sim A) \to A) \to A)$. Prin aplicarea succesivă a regulii MP : $\underset{L^*}{\vdash} A$, ceea ce contrazice ipoteza că A nu este o teoremă a lui L^*.

Scanned with OKEN Scanner

L nu este complet.

Demonstrație. Oricare din atomii p_i este o formulă, dar nici p_i nici $\sim p_i$ nu sînt teoreme în L.

Orice extensie inconsistentă L' a lui L este completă : L^c.

Demonstrație. În cazul unei extensii inconsistente L', orice formulă A este o teoremă a lui L' și astfel definiția unui sistem complet este satisfăcută.

Dacă L^c este extensie consistentă completă a lui L, orice altă extensie $L^{c'}$ a sa este inconsistentă.

Demonstrație. Fie A o formulă care nu este teoremă a lui L^c. Atunci $\underset{L^c}{\vdash} \sim A$. Într-o extensie nouă aceasta ar trebui să fie tot o teoremă : $\underset{L^{c'}}{\vdash} \sim A$. Dar în extensia nouă A este teoremă : $\underset{L^{c'}}{\vdash} A$. Rezultă că $L^{c'}$ este inconsistent.

Fie L^* o extensie consistentă a lui L. Există o extensie consistentă completă L^{*c} a lui L^*.

Mulțimea formulelor lui L poate fi scrisă sub forma șirului $A_1, A_2, \ldots, A_n, \ldots$ conform *principiului enumerării*, o proprietate generală a limbajelor infinite constînd în posibilitatea enumerării tuturor cuvintelor limbajelor într-un șir $w_1, \ldots, w_n, \ldots$ Într-adevăr, putem concepe enumerarea în felul următor : așezăm formulele lui L într-un șir în ordinea lungimii lor crescătoare, iar formulele de aceeași lungime le ordonăm după un criteriu similar cu al dicționarului. Construim apoi șirul de extensii ale lui L :

$$M_0 = L^*$$

$$M_1 = \begin{cases} M_0, \text{ dacă } \vdash A_0, \\ M_0 \cup \{\sim A_0\} \text{ altfel.} \end{cases}$$

Cu alte cuvinte, dacă A_0 este teoremă în M_0, atunci $M_1 = M_0$; dacă nu este, M_1 se formează adăugînd pe $\sim A_0$ ca o nouă axiomă a lui M_0. În general

$$M_{n+1} = \begin{cases} M_n, \text{ dacă } \underset{M_n}{\vdash} A_n, \\ M_n \cup \{\sim A_n\} \text{ altfel.} \end{cases}$$

Sistemele M_i sînt consistente. Procedăm prin inducție. Sistemul M_0 este consistent pentru că L^* este consistent. Dacă M_n este consistent, M_{n+1} este consistent, deoarece este construit în condițiile unei propoziții demonstrate mai înainte.

Demonstrație. Fie acum M o extensie a lui L^* care are ca axiome toate formulele care sînt axiome în cel puțin unul din sistemele șirului M_i. Sistemul M este consistent. Dacă ar fi inconsistent, ar exista o formulă B astfel ca $\underset{M}{\vdash} B$ și $\underset{M}{\vdash} \sim B$. Demonstrațiile lui B și $\sim B$, ca teoreme în M, sînt secvențe finite, deci cuprind un număr finit de axiome ale lui M. Pentru un n suficient de mare toate aceste axiome sînt axiomele lui M_n. Rezultă $\underset{M_n}{\vdash} B$ și $\underset{M_n}{\vdash} \sim B$, ceea ce contrazice consistența lui M_n.

Sistemul M este complet. Fie A o formulă oarecare a lui L. Ea trebuie să se găsească în enumerarea $A_1, \ldots, A_n, \ldots$ Presupunem că $A = A_k$.

Scanned with OKEN Scanner

Dacă nu avem $\underset{M_k}{\vdash} A_k$, atunci $\underset{M}{\vdash} \sim A_k$, sistemul M fiind o extensie a lui M_k, și $\sim A_k$ este axiomă a lui M_{k+1}, deci $\underset{M_{k+1}}{\vdash} \sim A_k$.

În acest caz, M fiind o extensie a lui M_{k+1}, avem $\underset{M}{\vdash} \sim A_k$. Concluzia : M este complet pentru că $\underset{M}{\vdash} A$ sau $\underset{M}{\vdash} \sim A$.

Dacă L^* este o extensie consistentă a lui L, există o evaluare v, pentru care $v(A) = 1$, oricare ar fi teorema A a lui L^*.

Demonstrație. Dacă am fi stabilit că orice teoremă este tautologie, enunțul ar fi trivial. Dar tocmai acest lucru ne pregătim să-l arătăm. Fie

$$v(A) = \begin{cases} 1, \text{ dacă } \underset{M}{\vdash} A, \\ 0, \text{ dacă } \underset{M}{\vdash} \sim A, \end{cases}$$

unde M este o extensie consistentă și completă a lui L. Întrucit M este complet, funcția v este definită pe mulțimea tuturor formulelor. Să arătăm că v este o evaluare, în sensul definiției date.

În primul rînd, $v(A) \neq v(\sim A)$, pentru că sistemul M este consistent. În al doilea rînd, să arătăm că $v(A \to B) = 0$, numai cînd $v(A) = 1$ și $v(B) = 0$.

Presupunem că $v(A) = 1$, $v(B) = 0$, dar $v(A \to B) = 1$. Atunci $\underset{M}{\vdash} A$, $\underset{M}{\vdash} \sim B$ și $\underset{M}{\vdash} (A \to B)$. Prin MP deducem $\underset{M}{\vdash} B$ și M nu ar fi consistentă. Concluzia : cînd $v(A) = 1$ și $v(B) = 0$, atunci $v(A \to B) = 0$.

Invers, presupunem că $v(A \to B) = 0$ și $v(A) = 0$ sau $v(B) = 1$. Atunci $\underset{M}{\vdash} \sim (A \to B)$ și $\underset{M}{\vdash} \sim A$ sau $\underset{M}{\vdash} B$. Pornim de la teoreme stabilite în L, la care adăugăm presupunerile

$$\underset{M}{\vdash} (\sim A \to (\sim B \to \sim A)), \qquad \underset{M}{\vdash} (B \to (A \to B))$$

$$\underset{M}{\vdash} \sim A \text{ sau } \underset{M}{\vdash} B$$

Prin MP : $\underset{M}{\vdash} (\sim B \to \sim A)$, $\underset{M}{\vdash} (A \to B)$,

$$\underset{M}{\vdash} ((\sim B \to \sim A) \to (A \to B)$$

Prin MP : $$\underset{M}{\vdash} (A \to B).$$

În ambele cazuri $\underset{M}{\vdash} (A \to B)$, ceea ce contrazice consistența lui M. Concluzionăm că v este o evaluare pentru $v(A \to B) = 0$ dacă și numai dacă $v(A) = 1, v(B) = 0$.

Fie o teoremă a lui L^*. Atunci $\underset{M}{\vdash} A$ pentru că M este o extensie a lui L^*. Urmează că $v(A) = 1$.

Dacă o formulă A a lui L este o tautologie, atunci este și teoremă în L.

Demonstrație. Dacă A (formulă și tautologie) n-ar fi teoremă în L, am putea construi extensia consistentă L^* obținută prin adăugarea lui $\sim A$ ca axiomă. Atunci există evaluarea v pentru care oricare teoremă are valoarea 1 și deci $v(\sim A) = 1$. Dar A fiind tautologie, $v(A) = 1$. Contradicție.

Scanned with OKEN Scanner

Comentariu asupra consistenței și completitudinii sistemelor formale.

a) Importanța consistenței unui sistem formal este capitală. Acest lucru se vede din faptul că într-un sistem inconsistent orice formulă este teoremă. Un sistem inconsistent conține contradicția, care implică orice, fapt care anihilează utilitatea sistemului. Un sistem în care orice formulă este teoremă este tot atît de inutil ca un sistem care nu are nici o teoremă.

b) Dintr-un sistem consistent, prin extensie, se pot obține alte sisteme consistente, capabile să demonstreze un număr mai mare de teoreme decît în sistemul inițial. Procedeul constă în transformarea unei formule a vechiului sistem într-o axiomă a noului sistem. Ca să se extindă, un sistem trebuie deci să aibă formule care să nu fie teoreme. Rezultă că extensia este posibilă atît timp cît sistemul nu este complet.

c) Condiția suficientă pentru consistență este foarte slabă: să existe o singură formulă care nu este teoremă. Într-un sistem consistent există multe formule care nu sînt teoreme; de exemplu negațiile tuturor teoremelor sale.

d) Procedeul extensiei se poate aplica repetat, incluzînd axiome adiționale și menținînd consistența, pînă la un punct și anume cînd sistemul devine complet. Extinzîndu-l pe acesta, obținem un sistem inconsistent.

Putem reprezenta combinațiile de mai înainte în fig. 2.1.

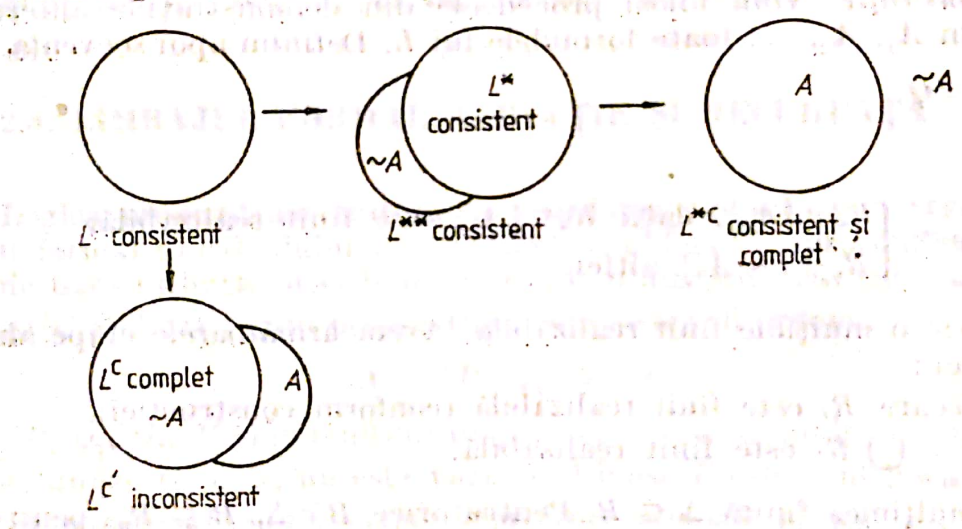


Fig. 2.1.

c) Faptul că orice teoremă este tautologie și orice tautologie este teoremă stabilește proprietatea importantă a sistemelor formale și anume că formulele demonstrabile în cadrul lor sînt tocmai formulele întotdeauna adevărate. Se realizează astfel o suprapunere perfectă a sintaxei peste semantică și echivalența demonstrabilității cu adevărul logic.

2.7. DECIDABILITATE ȘI COMPACTITATE

Un sistem formal L este *decidabil*, dacă există un procedeu efectiv pentru a decide dacă o formulă dată a lui L este sau nu teoremă în L.

Sistemul L al calculului propozițiilor este decidabil.

Scanned with OKEN Scanner

Într-adevăr, fie A o formulă. Dacă A este teoremă, este și tautologie. Stabilim că este tautologie recurgînd la tabla de adevăr, care conține toate evaluările v, care trebuie să dea $v(A) = 1$.

Am definit pînă acum tautologia ca o formulă adevărată pentru toate evaluările $v(A) = 1$, oricare ar fi v.

O formulă A este *realizabilă* cînd există cel puțin o evaluare v, pentru care $v(A) = 1$.

O mulțime R de formule este *realizabilă*, dacă există o evaluare v, astfel ca $v(A) = 1$, pentru toate $A \in R$.

O mulțime Q de formule este *finit realizabilă*, dacă orice mulțime finită Δ, $\Delta \subseteq Q$, este realizabilă.

Dacă Q este finit realizabilă, atunci $Q \cup \{A\}$ sau $Q \cup \{\sim A\}$ este finit realizabilă, unde A este o formulă oarecare, $A \notin Q$.

Demonstrație. Presupunem că nici una din mulțimile $Q \cup \{A\}$ și $Q \cup \{\sim A\}$ nu ar fi finit realizabilă. Atunci ar exista mulțimile finite $\Delta_1 \cup \{A\}$ și $\Delta_2 \cup \{\sim A\}$ nerealizabile. Dar $\Delta_1 \cup \Delta_2$ este realizabilă, fiind o submulțime finită a lui Q. Există deci evaluarea v, pentru care orice formulă $B \in \Delta_1 \cup \Delta_2$ are $v(B) = 1$. Pentru această evaluare sau $v(A) = 1$ sau $v(\sim A) = 1$. În primul caz $\Delta_1 \cup \{A\}$ este realizabilă, în cazul al doilea $\Delta_2 \cup \{A\}$ este realizabilă. Contradicție.

Orice mulțime finit realizabilă de formule este realizabilă.

Demonstrație. Vom folosi procedeele din demonstrațiile anterioare. Enumerăm $A_1, A_2, \ldots$ toate formulele lui L. Definim apoi secvența

$$R_0 = Q$$

$$\vdots$$

$$R_{n+1} = \begin{cases} R_n \cup \{A_n\}, & \text{dacă } R_n \cup \{A_n\} \text{ este finit realizabilă,} \\ R_n \cup \{\sim A_n\} & \text{altfel,} \end{cases}$$

unde Q este o mulțime finit realizabilă. Avem următoarele etape ale demonstrației:

a) Fiecare R_i este finit realizabilă (conform construcției).

b) $R = \bigcup_i R_i$ este finit realizabilă.

Fie mulțimea finită $\Delta \subseteq R$. Pentru orice $B \in \Delta$, $B \subseteq R_n$, pentru un n anumit. Deci $\Delta \subseteq R_m$, unde $m = \min n$. Dar R_m este finit realizabilă și deci Δ este realizabilă; ca urmare R este finit realizabilă.

c) Oricare ar fi formula A, avem sau $A \in R$ sau $\sim A \in R$.

Fie $A = A_n$, deci $A \in R_{n+1}$ sau $\sim A \in R_{n+1}$ și deci $A \in R$ sau $\sim A \in R$. Dacă amîndouă ar aparține lui R, mulțimea finită $\Delta = \{A, \sim A\}$ ar trebui să fie realizabilă, ceea ce este imposibil.

d) Oricare ar fi formula A, $v(A) = 1$ dacă și numai dacă $A \in R$.

În acest enunț evaluarea v este definită astfel: $v(p) = 1$, dacă $p \in R$ și $v(p) = 0$, dacă $p \neq R$, p fiind oricare atom.

Vom proceda prin inducție. Știm că A se exprimă cu conectorii $\sim$, $\rightarrow$. Dacă numărul acestora este 0, formula A este un atom. Deci propoziția este satisfăcută. Dacă A nu este atom, este de forma $\sim B$ sau $B \rightarrow C$, unde B și C, prin ipoteză, satisfac $v(B) = 1$, $v(C) = 1$, dacă și numai dacă B, $C \in R$.

Scanned with OKEN Scanner

Cazul $A = \sim B$. Folosind propoziția c), avem $v(A) = 1$ dacă și numai dacă $v(B) = 0$, dacă și numai dacă $B \notin R$, dacă și numai dacă $A \in R$. În cazul $A = B \to C$, demonstrația $v(A) = 1 \Leftrightarrow A \in R$ se bazează pe definiția evaluării v și pe proprietatea lui R, că toate submulțimile sale finite să fie realizabile.

Cu această pregătire se poate demonstra propoziția principală:

Fie Q o mulțime de formule finit realizabilă. Atunci $Q \subseteq R$, deci există o evaluare v pentru care $v(A) = 1$, oricare ar fi $A \in Q$. Atunci Q este realizabilă.

Model. Fie o mulțime de formule Q ale lui L. O evaluare în care fiecare element al lui Q este o evaluare adevărată, $v(A)=1$, $\forall A$, $A \in Q$, se numește *model* al lui Q.

Fie L un sistem formal. Un model al sistemului este evaluarea, în care fiecare teoremă este adevărată.

Propoziția demonstrată, conform căreia în extensia L^* consistentă există o evaluare astfel ca pentru orice teoremă A din L să avem $v(A) = 1$ se poate enunța astfel:

Sistemul L^* este consistent dacă are un model.

Iar teorema compactității se poate enunța:

Dacă orice submulțime finită de formule ale unui sistem consistent al calculului propozițiilor are model, sistemul are model.

2.8. LIMBAJUL FORMAL. INDUCȚIE ȘI RECURENȚĂ

Logica propozițiilor tratată în abordarea sintactică în studiul sistemului formal al calculului propozițiilor ne obligă să aprofundăm conceptele de bază utilizate și să le definim într-o manieră mai riguroasă.

Alfabetul calculului propozițiilor este sistemul ordonat

$$A = \{P,\ Q_1,\ Q_2,\ U\},$$

unde P, Q_1, Q_2, U sînt mulțimi disjuncte, P este o mulțime numărabilă, iar reuniunea $Q_1 \cup Q_2$ nu este vidă. Mulțimea P este mulțimea simbolurilor propoziționale sau a variabilelor propoziționale $p_1, p_2, \ldots$, mulțimea Q_1 este mulțimea conectorilor unari, iar Q_2 mulțimea conectorilor binari. Mulțimea U este mulțimea simbolurilor auxiliare, Q_1 cuprinde numai simbolul $\sim$, iar Q_2 simbolul $\to$. În alte sisteme formale Q_2 conține conectorii binari $\vee$, $\wedge$, $\to$. Mulțimea U conține parantezele (și).

Alfabet de ordinul 0 este alfabetul care nu conține alte simboluri decît cele menționate. *Alfabetul de ordinul* 1 care va fi utilizat în calculul cu predicate conține simboluri pentru predicate, relații, variabile și cuantificatori.

Presupunem că nici un simbol al alfabetului nu este o secvență finită de alte simboluri. Aceasta are drept consecință faptul că simbolurile sînt distincte. Astfel:

$$p_4 \neq \langle \sim, p_2, \to \rangle.$$

Scanned with OKEN Scanner

Altă consecință este proprietatea secvențelor finite de a fi decompozabile în mod unic. Dacă

$$\langle a_1, \ldots, a_m \rangle = \langle b_1, \ldots, b_n \rangle,$$

unde prin $\langle\ \rangle$ se notează o secvență, și dacă fiecare a_i, b_j, $i = 1, \ldots, n$, $j = 1, \ldots, m$, sînt simboluri, atunci $m = n$ și $a_i = b_i$.

O secvență finită de simboluri se numește *expresie*. O expresie poate fi scrisă prin concatenare de simboluri; astfel expresia $(\sim p_1)$ este secvența $\langle (, \sim, p_1,) \rangle$. În general, dacă α și β sînt secvențe de simboluri, $\alpha\beta$ este secvența formată din simbolurile secvenței α urmate de simbolurile secvenței β. Astfel $(\alpha \to\ )$, unde $\alpha = (\sim p_1)$ și $\beta = p_2$, este expresia $((\sim p_1) \to p_2)$.

Calculul propozițiilor face o delimitare strictă între expresii și formule, primele putînd fi secvențe „fără sens", neutilizabile. Calculul propozițiilor operează numai asupra formulelor.

Alfabetul și formulele constituie limbajul calculului propozițional. Perechea $\mathscr{L} = \{A, F\}$, unde A este un alfabet de ordin 0 și F este mulțimea formulelor formate cu simboluri în A, este numit *limbaj formalizat de ordinul* 0 sau *limbaj formalizat al unui calcul propozițional*.

Constituirea corectă a formulelor din simbolurile alfabetului cere ca definiția introdusă anterior (v. 2.1) să fie analizată. În cele ce urmează cuvîntul „formulă" va desemna formulă bine formulată (fbf).

Introducem noțiunea de *mulțime inductivă*. O mulțime de expresii F este *inductivă* dacă are proprietățile:

1) fiecare simbol propozițional este în F, $P \subseteq F$,

2) dacă expresiile a și b sînt în F, atunci $(\sim a)$ și $(a \to b)$ sînt în F.

Definiția 1. O expresie este o *formulă* dacă și numai dacă aparține fiecărei mulțimi inductive.

Comparînd această definiție cu cea enunțată în 2.1, se vede că a) și b) sînt satisfăcute.

Cerința c) ca o formulă să nu fie obținută decît prin regulile a) și b) este echivalentă cu a cere ca mulțimea formulelor să fie submulțime a oricărei mulțimi inductive, adică cea mai mică posibilă. (Orice mulțime inductivă de formule este de fapt mulțimea tuturor formulelor. Această proprietate reflectă principiul inducției.)

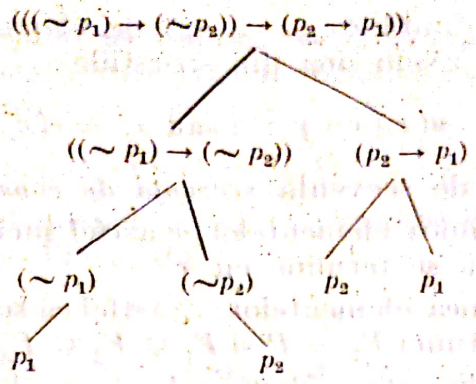
Definiția care introduce formula prin proprietatea apartenenței la oricare mulțime inductivă procedează „de sus în jos". Pentru a satisface cerința c) putem acționa și „de jos în sus", introducînd operațiile de construire a formulelor, definite astfel pe mulțimea expresiilor:

$$O^{\sim}(\alpha) = (\sim\alpha), \quad O^{\to}(\alpha, \beta) = (\alpha \to \beta).$$

Definiția 2. O *expresie* este o formulă dacă și numai dacă poate fi construită din simboluri prin aplicarea de un număr finit de ori a operațiilor de construire a formulelor.

Scanned with OKEN Scanner

Exemplu. Formula $(((\sim p_1) \to (\sim p_2)) \to (p_2 \to p_1))$ poate fi analizată cu ajutorul unui arbore, format prin descompunerea în subformule a formulei date:



Formula este formată din două simboluri și din aplicarea de două ori a operației $O^{\sim}$ și de trei ori a operației $O^{\to}$.

Inducție. Construirea inductivă a formulelor ilustrează metoda generală a inducției, larg folosită în logică și constînd din construirea unei submulțimi $B \subseteq A$, pornind de la o anumită submulțime C și aplicînd anumite operații în mod repetat. Mulțimea B pe care o căutăm este cea mai mică submulțime care conține elementele inițiale din C și care este închisă față de operațiile definite pe A. Membrii ei vor fi acele elemente ale lui A care pot fi construite din elementele lui C prin aplicarea operațiilor de un număr finit de ori.

În cazul examinat fie E mulțimea expresiilor, $P \subseteq E$ elementele inițiale care sînt simboluri propoziționale și operațiile $O^{\sim}$ și $O^{\to}$. Urmărim construirea submulțimii F a formulelor.

Pornim de la $F \subseteq E$ și de la o clasă $\mathcal{F}$ de funcții ce corespund operațiilor $\sim$ și $\to$ și anume:

$$n : E \to E, \qquad c : E \times E \to E,$$

unde n este o funcție unară ce corespunde negației, iar c o funcție binară ce corespunde implicației. (În notația Lukasiewicz, negația este notată N, iar implicația prin C.) A spune că cele două operații creează dintr-un element sau o pereche de elemente noi elemente ale mulțimii E este echivalent cu a considera aplicațiile $E \to E$ și $E \times E \to E$.

Presupunem că P este format din două elemente inițiale p_1 și p_2. Din aceste elemente, prin acțiunea celor două funcții se obțin noi elemente de forma

$$c(p_1, p_2),\ n(p_1),\ c(n(p_1),\ c(p_1, p_2)),\ n(c(n(p_2),\ c(p_1, p_1))) \text{ etc.}$$

Pentru definirea mulțimii formulelor F în mod formal avem cele două posibilități semnalate. Prima este calea „de sus în jos" caracterizată de următoarea serie de definiții:

a) $S \subseteq E$ este închisă față de n și c, dacă $n(x)$ și $c(x, y)$ aparțin lui S, oricare ar fi $x, y \in S$.

b) S este inductivă dacă și numai dacă $P \subseteq S$ și S este închisă față de n și c.

c) Fie F^* intersecția tuturor submulțimilor inductive S ale lui E: $x \in F^*$, dacă x aparține fiecărei submulțimi inductive.

Se demonstrează că F^* este ea însăși inductivă și că este cea mai mică submulțime inductivă inclusă în toate submulțimile inductive. Am obținut astfel mulțimea formulelor F^*.

Scanned with OKEN Scanner

A doua posibilitate „de jos în sus" este caracterizată de următoarele definiții :

a') Fie secvența finită $\langle x_0, \ldots, x_m \rangle$ de elemente din E, astfel ca pentru fiecare $i \leqslant m$ avem una din situațiile :

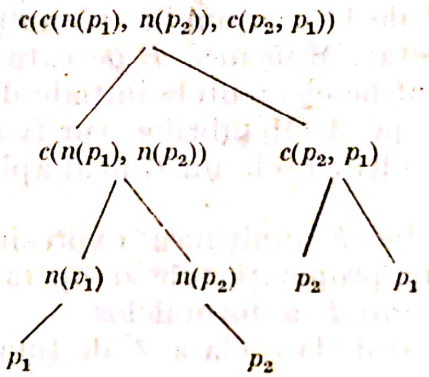
$$x_i \in P \text{ sau } x_i = n(x_j) \text{ cu } j < i \text{ sau } x_i = c(x_j, x_k) \text{ cu } j, k < i.$$

Numim o astfel de secvență *secvență de construcție*.

b') Fie F_* mulțimea elementelor x astfel încît să existe o secvență de construcție care să se termine cu x.

c') Fie F_m mulțimea elementelor x, astfel ca secvențele de lungime m să se termine cu x. Atunci $F_1 = P$ și $F_1 \subseteq F_2 \subseteq F_3$, iar F_* este reuniunea mulțimilor F_i. Aceasta este a doua cale de a obține mulțimea tuturor formulelor.

Exemplu. Reluăm formula din exemplul precedent :



În acest caz formula se află în F_5 și deci în F_*.

Să arătăm că cele două definiții sînt echivalente și că $F = F^* = F_*$.

1° $F^* \subseteq F_*$. Este suficient să arătăm că F_* este inductivă, adică $P \subseteq F_*$ și că F_* este închisă față de n și c. Dar $P = F_1 \subseteq F$. Dacă x și y sînt în F_*, concatenăm secvențele de construcție și atașăm $c(x, y)$ pentru a obține secvența de construcție care plasează pe $c(x, y)$ în F_*. La fel, F_* este închisă față de n.

2° $F_* \subseteq F^*$. Considerăm un element al lui F_* și secvența sa de construcție $\langle x_0, \ldots, x_m \rangle$. Prin inducție după i, se demonstrează că $x_i \in F^*$, $i \leqslant m$: $x_0 \in P \subseteq F^*$, iar pentru trecerea de la i la $i + 1$ folosim faptul că F^* este închisă față de n și c.

Pentru că definițiile sînt echivalente și $F^* = F_*$ putem să numim $F = F^* = F_*$ mulțimea formulelor ca o mulțime generată de P prin funcțiile n, c. Mulțimea formulelor poate fi astfel concepută ca intersecția tuturor mulțimilor inductive sau ca reuniunea tuturor mulțimilor de elemente ce pot fi obținute prin secvențe finite de construcție.

În general, principiul ilustrat în cazul construirii formulelor unui limbaj se enunță astfel :

Principiul inducției. *Fie B mulțimea generată de C prin funcțiile $\mathscr{F}$. Dacă S este o submulțime a lui B care include pe C și este închisă față de funcțiile din $\mathscr{F}$, atunci $S = B$.*

Acesta nu este singurul principiu al inducției, folosit în demonstrații. Există demonstrații prin inducție după lungimea expresiilor, după numărul

de locuri în care apare un conector, după indicele de enumerare a unei mulțimi etc.

Interpretarea limbajului de ordinul 0. O *interpretare* a unui limbaj de ordinul 0 este o aplicație care atribuie fiecărei variabile propoziționale o valoare de adevăr 0 sau 1 :

$$v : P \to \{0, 1\}.$$

Această evaluare a simbolurilor $p_1, p_2, \ldots$ trebuie extinsă la o evaluare pentru toate formulele limbajului :

$$\bar{v} : F \to \{0, 1\}.$$

În acest scop au fost introduse reguli, care permit să calculăm pe $\bar{v}(x)$ cînd $x \in P$, pe $\bar{v}(n(x))$ utilizînd pe $\bar{v}(x)$ și pe $\bar{v}(c(x, y))$ utilizînd pe $\bar{v}(x)$ și $\bar{v}(y)$. Definiția recursivă a lui $\bar{v}$ este :

$$\bar{v}(x) = v(x), \text{ dacă } x \in P,$$

$$\bar{v}(n(x)) = \begin{cases} 1, & \text{dacă } \bar{v}(x) = 0, \\ 0 & \text{altfel,} \end{cases}$$

$$\bar{v}(c(x, y)) = \begin{cases} 0, & \text{dacă } \bar{v}(x) = 1 \text{ și } \bar{v}(y) = 0, \\ 1 & \text{altfel.} \end{cases}$$

Extensia evaluării v la $\bar{v}$ poate fi enunțată în termeni mai generali astfel : fiind date mulțimea E, $P \subseteq E$, două funcții

$$n : E \to E, \qquad c : E \times E \to E$$

și F mulțimea generată de P prin n și c, cum definim recursiv o funcție pe F (în cazul de față $\bar{v}$) ?

Răspunsul la această întrebare este dat de *teorema recurenței*, care presupune însă o întărire a cerinței ca F să fie generat de P prin n și c și anume :

Definiție. Mulțimea F este *generată liber* de P prin n și c, dacă și numai dacă, în afara condițiilor de generare, avem

1° n_F și c_F (restricțiile funcțiilor n și c la F) sînt aplicații injective :

2° imaginea lui n_F, imaginea lui c_F și mulțimea P sînt disjuncte două cîte două.

Reamintim noțiunile din teoria mulțimilor.

O *funcție* $f : A \to B$ aplică mulțimea A pe mulțimea B sau este definită pe A (domeniul lui f, $\mathrm{dom} f$) dacă fiecărui element din A îi asociază un singur element din B. Se numește *imaginea lui* f, $f(A)$, notată $\mathrm{Im}(f) = f(A) \subset B$. Cînd $\mathrm{Im}(f) = B$, aplicația este *surjectivă* (v. 2.2, a) ; cînd sînt aplicate elemente distincte din A în elemente distincte din B, $x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$, aplicația este *injectivă* (2.2, b) ; cînd aplicația este surjectivă și injectivă se numește *bijecție* (v. 2.2, c, exemplu în fig. 2.2).

Funcția $f_{A'}$ este *restricția lui* f față de o submulțime A' a domeniului lui f, dacă și numai dacă domeniul său este o submulțime a domeniului lui f și $f_{A'}(x) = f(x)$. Complementarea noțiunii de restricție este extensia unei funcții : spunem că f este o *extensie* a lui $f_{A'}$. Un *omomorfism* h, $h : (A, o_1) \to (B, o_2)$, este o aplicație care păstrează operațiile : $h(o_1(x)) = o_2(h(x))$. Un omomorfism injectiv este *izomorfism*.

Scanned with OKEN Scanner

Teorema recurenței. Dacă presupunem că submulțimea $F \subseteq E$ este generată liber de F prin n și c, unde $n : E \to E$ și $c : E \times E \to E$, și că B este o mulțime, iar N și C sînt funcții astfel ca $v : P \to B$, $N : B \to B$,

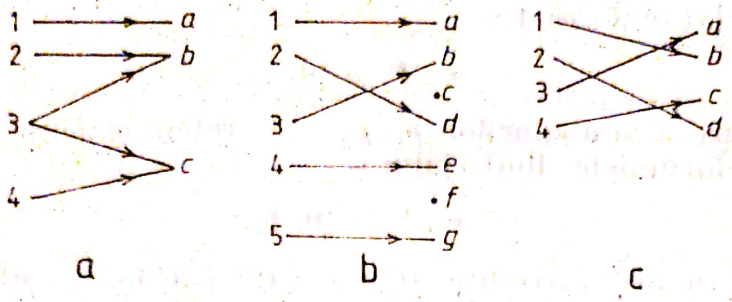


Fig. 2.2.

$C : B \times B \to B$, atunci există o funcție unică $\bar{v} : F \to B$, astfel încît:

a) pentru $x \in P$, $\bar{v}(x) = v(x)$,

b) pentru $x \in F$, $\bar{v}(n(x)) = N(\bar{v}(x))$,

pentru $x, y \in F$, $\bar{v}(c(x, y)) = C(\bar{v}(x), \bar{v}(y))$,

(E, F, n, c, P au sensul atribuit mai sus, B este mulțimea $\{0, 1\}$, iar N și C sînt operațiile $\sim$ și $\to$ definite pe elementele acestei mulțimi).

Demonstrația se bazează pe posibilitatea concepерii lui $\bar{v}$ ca o reuniune de funcții de aproximare. În acest scop introducem funcțiile acceptabile care aplică o parte a lui F în B și satisfac condițiile impuse de a) și b) lui $\bar{v}$.

Definiție. O *funcție acceptabilă* f are ca domeniu o submulțime a lui F și ca imagine o submulțime a lui B și îndeplinește condițiile:

a′) dacă $x \in P$ și $x \in \operatorname{dom} f$, $f(x) = v(x)$,

b′) dacă $n(x) \in \operatorname{dom} f$, atunci și x și $f(n(x)) = N(f(x)) \in \operatorname{dom} f$. Dacă $c(x, y) \in \operatorname{dom} f$, atunci x, y și $f(c(x, y)) = C(f(x), f(y)) \in \operatorname{dom} f$.

Fie A mulțimea tuturor funcțiilor acceptabile și $\bar{v}$ reuniunea elementelor lui A. Vom avea $\langle x, y\rangle \in \bar{v}$ dacă și numai dacă $f(x) = y$ pentru un $f \in A$.

Funcția $\bar{v}$ va răspunde cerințelor formulate în teoremă. Etapele demonstrației sînt următoarele:

1° Se verifică dacă $\bar{v}$ este o funcție. Se ia mulțimea S formată din elementele x ale lui F, care satisfac condiția ca pentru cel mult un y are loc $\langle x, y\rangle \in \bar{v}$. Se verifică că S este inductivă, folosindu-se proprietățile a′) și b′). Atunci $S = F$ și $\bar{v}$ este o funcție.

2° Din definiția lui $\bar{v}$ și a proprietății că este o funcție rezultă că $\bar{v} \in A$, adică este acceptabilă.

3° Pentru a arăta că $\bar{v}$ este definită peste tot în F, este suficient să arătăm că domeniul lui $\bar{v}$ este inductiv. Presupunem că x aparține domeniului lui $\bar{v}$. Atunci, de exemplu $\langle n(x), N(\bar{v}(x))\rangle$ este acceptabilă. În consecință, $n(x)$ este în domeniul lui $\bar{v}$. În acest punct al demonstrației, ipoteza că F este liber generată de P a fost folosită pentru a arăta că $\langle n(x), N(\bar{v}(x))\rangle$ este acceptabilă. La fel pentru $c(x, y)$.

4° În sfîrșit, funcția $\bar{v}$ este unică. Fiind date două funcții, fie S mulțimea pe care ele iau valori egale. Mulțimea S este inductivă și deci $S = F$.

Prezentarea unei teoreme generale de algebră pentru cazul sistemului formal analizat constituie un prim contact cu algebra universală,

indispensabilă în aprofundarea logicii. În același timp este o pregătire pentru construirea modelelor algebrice ale sistemelor formale, dezvoltată în capitolul următor.

Din punct de vedere algebric, teorema recurenței afirmă că o aplicație $v : R \to B$ poate fi extinsă la un omomorfism $\bar{v} : F \to B$, operațiile în F fiind n și c, iar în B sînt N și C.

Condiția generării libere este esențială pentru demonstrarea teoremei. Cînd ea lipsește, existența lui $\bar{v}$ nu este asigurată. Acesta este cazul cînd încercăm în teoria grupurilor să extindem o aplicație $X \to B$ în omomorfismul $G \to B$, unde G este un grup generat de subgrupul X, iar B este grup. Rezultatul nu este în mod necesar posibil, în afara cazului cînd adăugăm condiția ca G să fie grup liber și X o mulțime de generatori independenți.

Aplicații. 1. O aplicație a teoremei este posibilitatea de a defini o funcție l pe mulțimea F care să satisfacă condițiile :

$$l(p) = 1 \text{ pentru orice simbol propozițional,}$$

$$l((\sim\alpha)) = 3 + l(\alpha)$$

$$l((\alpha \to \beta)) = 3 + l(\alpha) + l(\beta).$$

Aplicația l se numește *lungimea unei formule*.

2. Orice expresie cu mai multe paranteze stîngi decît drepte nu este o formulă. Demonstrația ilustrează aplicarea principiului inducției. Proprietatea de a avea același număr de paranteze stîngi și de paranteze drepte caracterizează o formulă „echilibrată". Mulțimea formulelor echilibrate conține pe P și este închisă față de operațiile de formare a formulelor. Este deci inductivă și conform principiului inducției este egală cu mulțimea formulelor.

3. Orice segment inițial propriu al unei formule conține mai multe paranteze stîngi decît drepte. Prin urmare, nici un segment inițial propriu al unei formule nu este el însuși formulă.

Un segment al unei secvențe finite $\langle x_1, \ldots, x_n \rangle$ este secvența $\langle x_k, x_{k+1}, \ldots, x_m \rangle$, unde $1 \leqslant k \leqslant m \leqslant n$. Dacă $k = 1$, segmentul este *inițial*. Dacă segmentul este diferit de secvența inițială, segmentul este *propriu*. Formăm iarăși o mulțime S a formulelor care au proprietatea ca segmentele lor inițiale să aibă mai multe paranteze stîngi decît drepte. Arătăm că S este inductivă și închisă față de operații.

De exemplu, pentru a arăta că mulțimea S este inductivă și închisă față de $\to$ considerăm șase segmente inițiale proprii : 1. $($; 2. $(\alpha_0$, unde α_0 este segment inițial propriu al lui α ; 3. α ; 4. $(\alpha :\to$ 5. $(\alpha \to \beta_0$, unde β_0 este segment inițial propriu al lui β și 6. $(\alpha \to \beta$.

4. Mulțimea formulelor este liber generată de mulțimea P și de $O^{\sim}$ și $O^{\to}$.

5. Următorul algoritm, bazat pe proprietățile enunțate în 1—4, permite construirea unui arbore unic asociat fiecărei formule. Fie dată o expresie. Ea este nodul superior al arborelui. Din acesta se coboară arce în alte noduri, obținîndu-se astfel un graf (arbore).

1) Procedura este încheiată cînd în nodurile minimale sînt simboluri propoziționale. Altfel continuă.

2) Primul simbol trebuie să fie $($. În caz contrar expresia nu este formulă. Dacă al doilea simbol este $\sim$, urmează pasul 4.

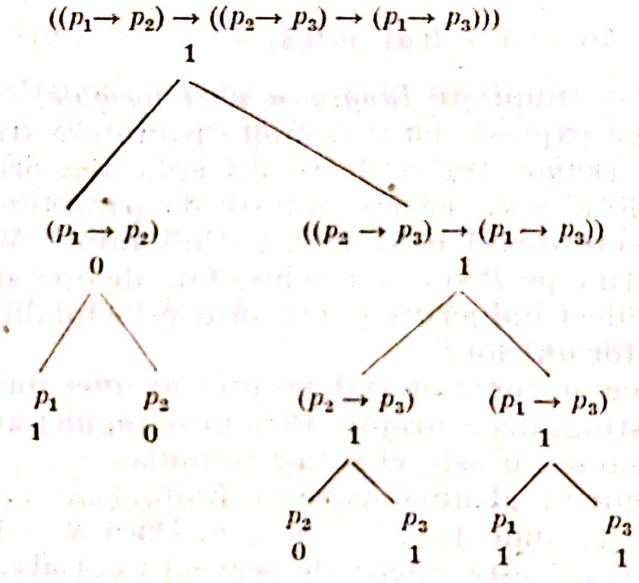
3) Se parcurge expresia de la stînga la dreapta pînă se întîlnește $(\alpha$, unde α este o expresie cu același număr de paranteze stîngi și drepte ;

α este primul constituent. Următorul simbol trebuie să fie $\rightarrow$; el este conectorul principal. În caz contrar expresia nu este formulă. Restul expresiei β) este o expresie urmată de o paranteză dreaptă). În caz contrar expresia nu este o formulă. Al doilea constituent este β. Expresia a fost descompusă. Revenire la punctul 1.

4) Dacă al doilea simbol este $\sim$, acesta este conectorul principal. Restul expresiei β) trebuie să consiste dintr-o expresie și o paranteză dreaptă. Altfel expresia nu este formulă; β este constituentul. Revenire la punctul 1.

Algoritmul scoate în evidență situația în care expresia dată nu este formulă. El se oprește după un număr finit de pași. Fiecare nod conține o expresie de o lungime mai mică decît precedentul. Arborele servește și la obținerea evaluării unei formule într-o interpretare dată, asociind valori nodurilor începînd cu cele minimale pînă la nodul superior.

Exemplu



Expresia dată este o formulă, cu arborele asociat care a rezultat și care servește la interpretarea $v(p_1) = 1$, $v(p_2) = 0$, $v(p_3) = 1$, pentru care formula este adevărată.

2.9. CALCULUL CU ARBORI DE DEDUCȚIE

Regula de inferență pe care o utilizează calculul propozițiilor este modus ponens (MP), care se enunță : din A și $A \rightarrow B$ rezultă B. Această regulă corespunde unei teoreme

$$\vdash A \rightarrow ((A \rightarrow B) \rightarrow B)$$

și se scrie sub formă de regulă :

$$\frac{A,\ A \rightarrow B}{B} \text{ sau } \frac{\vdash A,\ \vdash A \rightarrow B}{\vdash B}$$

Ideea de a scrie și alte teoreme sub formă de reguli și de a le folosi pe acestea într-un calcul pentru a obține noi teoreme a dus la crearea *calculului cu scheme* sau *calculului natural*. În calculul clasic cu propoziții (nu în mod necesar în alte logici) orice regulă poate fi transformată într-o teoremă. În acest sens s-a afirmat că „teoremele sînt reguli înghețate, iar regulile teoreme dezghețate".

Exemplu. Regulile

$$\frac{A}{B} \qquad \frac{A,\ B}{C} \qquad \frac{A \to B,\ B \to C}{A \to C}$$

corespund teoremelor

$$\vdash A \to B \qquad \vdash A \to (B \to C) \qquad \vdash (A \to B) \to ((B \to C) \to (A \to C)).$$

Din aceste teoreme, prin aplicarea MP rezultă afirmația exprimată de regulă. Este de observat că această afirmație nu este decît faptul că dacă acceptăm ca ipoteză ceea ce este la partea superioară a fracției, atunci rezultă consecința din partea inferioară.

Mai general, putem scrie $\frac{A}{B}$ și $\frac{A, B}{C}$ sub forma

$$\frac{A_1, \ldots, A_n \vdash A}{A_1, \ldots, A_n \vdash B} \text{ și } \frac{A_1, \ldots, A_n \vdash A,\ A_1, \ldots, A_n \vdash B}{A_1, \ldots, A_n \vdash C}$$

cu sensul: dacă A este derivabilă din $A_1, \ldots, A_n$, atunci B este derivabilă din $A_1, \ldots, A_n$ sau dacă A și B sînt derivabile din $A_1, \ldots, A_n$, atunci este și C.

Definiție. Numim *secvenți* ai calculului propozițiilor secvențele de următorele patru forme:

1) $A_1, \ldots, A_n \vdash A$; 2) $A_1, \ldots, A_n \vdash$; 3) $\vdash A$; 4) $\vdash$; unde $A, A_1, \ldots, A_n$ sînt formule.

Într-un secvent se pot distinge premisele de consecințe. Dacă atribuim lui $\vdash$ sensul de consecință logică, atunci putem înțelege forma 1) ca „validitatea (adevărul) premiselor (ipotezeler, condițiilor) $A_1, \ldots, A_n$ are drept consecință (urmare, rezultat) propoziția A", iar forma 2) „$A_1, A_2, \ldots, A_n$ sînt inconsistente".

Pentru introducerea unui calcul cu scheme, este nevoie să completăm alfabetul de simboluri al calculului propozițiilor cu simboluri pentru formule $A, B, C, \ldots$ sau $A_1, A_2, A_3, \ldots$ și cu simboluri pentru secvențe de formule $\alpha, \beta, \gamma, \ldots$

O *schemă de secvenți* este o expresie astfel ca orice substituție a simbolurilor $A, B, C, \ldots$ și $\alpha, \beta, \gamma, \ldots$ cu formule și secvenți să rezulte într-un secvent. Printr-o asemenea substituție rezultă o *instanță a schemei*. O *schemă de formule* este o expresie astfel ca orice substituție de simboluri să rezulte într-o formulă.

Scanned with OKEN Scanner

Exemplu. a) Schema de secvenți $\alpha,\ A \vdash A \vee B$ are ca instanță

$$A_1 \vee A_2,\ \sim A_3,\ A_1 \wedge A_3,\ A_1 \to A_2 \vdash (A_1 \to A_2) \vee (A_1 \to A_3).$$

b) Schema de formule $A \to (B \vee C)$ are ca instanță

$$(A_1 \to A_2) \to ((A_1 \to A_3) \vee (A_1 \wedge A_2 \wedge A_3)).$$

Axiomă este o instanță a schemei $A \vdash A$, numită *schemă axiomă*. *Regulile de inferență* sînt următoarele :

1) $\dfrac{\alpha \vdash A\ ;\ \alpha \vdash B}{\alpha \vdash A \wedge B}$

2) $\dfrac{\alpha \vdash A \wedge B}{\alpha \vdash A}$

3) $\dfrac{\alpha \vdash A \wedge B}{\alpha \vdash B}$

4) $\dfrac{\alpha \vdash A}{\alpha \vdash A \vee B}$

5) $\dfrac{\alpha \vdash B}{\alpha \vdash A \vee B}$

6) $\dfrac{\alpha, A \vdash B\ ;\ \alpha, C \vdash B\ ;\ \alpha \vdash A \vee C}{\alpha \vdash B}$

7) $\dfrac{\alpha, A \vdash B}{\alpha \vdash A \to B}$

8) $\dfrac{\alpha \vdash A\ ;\ \alpha \vdash A \to B}{\alpha \vdash B}$

9) $\dfrac{\alpha, \sim A \vdash}{\alpha \vdash A}$

10) $\dfrac{\alpha \vdash A\ ;\ \alpha \vdash \sim A}{\alpha \vdash}$

11) $\dfrac{\alpha, A, B, \beta \vdash C}{\alpha, B, A, \beta \vdash C}$

12) $\dfrac{\alpha \vdash A}{\alpha, B \vdash A}$

Se observă că regulile incorporează diferitele moduri de a infera. Acestea pot fi înțelese astfel : 1) dacă din secvența α rezultă atît A cît și B, rezultă și conjuncția lor ; 2) și 3) dacă rezultă o conjuncție, rezultă fiecare din membrii conjuncției ; 4) și 5) dacă din α rezultă A, rezultă și conjuncția lui cu o formulă oarecare B ; 6) dacă B rezultă din α și A, și din α și C, iar din α rezultă A sau C, atunci B rezultă din α (se mai numește *raționamentul celor două cazuri)*. Regula 7) corespunde teoremei deducției. Regula 8) este regula detașării (modus ponens). Regula 9), din faptul că α și $\sim A$ este inconsistentă, rezultă A (se numește *regula reducerii la absurd*) ; 10) dacă din α rezultă atît A, cît și $\sim A$, rezultă că α este inconsistentă (se numește și *regula contradicției*) ; 11) schimbarea ordinii unor ipoteze nu afectează concluzia ; 12) adăugarea unei premise nu afectează concluzia (se numește și *proprietatea de monotonie, regula de rafinare* sau *regula ipotezei suplimentare)*.

Din axiome prin aplicarea repetată a regulilor urmărim să obținem noi scheme. Următorul exemplu va ușura înțelegerea formalizării procedeului. Diagrama reprezintă obținerea schemei $A,\ B \vdash A \wedge B$ din axio-

mele $A \vdash A$, $B \vdash B$ și regulile 12), 11), 1) scrise în dreptul liniilor

$$\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,(12) \quad \dfrac{\dfrac{B \vdash B}{B, A \vdash B}\,(12)}{A, B \vdash B}\,(11)}{A, B \vdash A \wedge B}\,(1)$$

Definiții. 1) O *demonstrație liniară* este secvența finită de secvenți $S_1, \ldots, S_n$ astfel ca fiecare secvent S_i $(i \leqslant n)$ este sau axiomă sau se obține din S_j $(j < i)$ prin regulile 1—12. În exemplul nostru $B \vdash B$; $B, A \vdash B$ și $A, B \vdash B$ este o demonstrație liniară.

2) Un secvent este *demonstrabil* sau este *teoremă* dacă există o demonstrație liniară $S_1, \ldots, S_n$ astfel ca $S_n = S$.

3) O formulă A este demonstrabilă dacă secventul $\vdash A$ este demonstrabil.

4) Un *arbore* T este definit recursiv:

a) orice secvent este arbore;

b) dacă $T_1, \ldots, T_n$ sînt arbori, atunci $\dfrac{T_1, \ldots, T_n}{S}$ este arbore.

În exemplul de mai sus, toate expresiile conținînd simbolul $\vdash$ (secvenții) sînt arbori ca și toate diagramele avînd linia orizontală.

5) Un secvent este *inițial* în T cînd nu are linie deasupra lui în arbore. De exemplu: $A \vdash A$. Un secvent este *final* în T cînd nu are linie sub el, ca de exemplu: $A, B \vdash A \wedge B$.

6) Partea unui arbore constînd dintr-o linie și secvenții de deasupra și dedesubtul liniei se numește *trecere*. Arborele din exemplu are patru treceri.

7) Un *arbore de demonstrație* este un arbore ai cărui secvenți inițiali sînt axiome, iar trecerile aplicații ale regulilor 1—12 (este cazul exemplului dat).

8) Dacă S este secventul final al lui T, spunem că T este arborele de demonstrație al lui S sau *arborele de derivare* al lui S.

9) Se numește *înălțimea unui secvent* într-un arbore T funcția definită astfel:

a) $h(S) = 0$, dacă S este secvent final în T,

b) $h(S_1) = \ldots = h(S_n) = h(S) + 1$ dacă $\dfrac{S_1, \ldots, S_n}{S}$ este o trecere în T.

10) *Înălțimea arborelui* este înălțimea maximă a secvenților arborelui.

Cu definițiile 1—10 putem demonstra următoarea

Propoziție. *Un secvent S are un arbore de demonstrație numai și numai dacă S este teoremă.*

Demonstrație. Dacă S este teoremă, există o demonstrație liniară $S_1, \ldots, S_n = S$. Dacă S este axiomă, S este arborele de demonstrație. Fie $T_1, \ldots, T_{n-1}$ arborii de demonstrație ai secvenților $S_1, \ldots, S_{n-1}$.

Scanned with OKEN Scanner

Dacă $S_{i_1}, \ldots, S_{i_k}$, cu $i_1, \ldots, i_k < n$ este aplicarea unei reguli, atunci arborele

$$\frac{T_{i_1}, \ldots, T_{i_k}}{S}$$

este arborele de demonstrație al lui S.

Invers, dacă S are un arbore de demonstrare T, construim o demonstrație liniară prin inducție după înălțimea secvenților din arborele T. Secvenții inițiali sînt demonstrații liniare. Dacă toți secvenții $S_1, \ldots, S_m$ de înălțime de lungime $k+1$ au demonstrații liniare $D_1, \ldots, D_m$, secvența $D_1, \ldots, D_m, S$ va fi o demonstrație liniară pentru S cu înălțimea k.

Continuăm lista definițiilor :

11) O schemă de secvenți este *demonstrabilă* dacă adăugînd-o ca o schemă axiomă nu lărgim mulțimea secvenților demonstrabili. Aceasta înseamnă că toate instanțele schemei demonstrabile sînt demonstrabile.

12) O regulă de inferență este *admisibilă* dacă adăugînd-o calculului propozițiilor nu lărgim mulțimea secvenților demonstrabili. Regulile 1—12 sînt admisibile.

13) O *cvasiderivație a secventului* S este o secvență finită de secvenți $S_1, \ldots, S_n = S$ dacă fiecare secvent ce figurează în ea este demonstrabil sau poate fi obținut din secvenții precedenți prin reguli admisibile de inferență. Observăm că, față de definiția demonstrabilității care cere aplicarea regulilor 1—12, cvasiderivația face uz de orice regulă admisibilă de inferență.

14) Un arbore T este *arbore de cvasiderivație* al unui secvent S dacă orice secvent inițial este demonstrabil (nu neapărat axiomă), secventul final este S și trecerile sînt reguli admisibile (nu neapărat 1—12, ca în cazul derivării).

Un secvent pentru care există o cvasiderivație sau un arbore de cvasiderivație este demonstrabil.

Reguli admisibile sînt următoarele :

1′) $\dfrac{A_1, \ldots, A_n \vdash A}{B_1, \ldots, B_m \vdash A}$, unde $\{A_1, \ldots, A_n\} \subseteq \{B_1, \ldots, B_m\}$,

2′) $\dfrac{A_1, \ldots, A_n \vdash}{B_1, \ldots, B_m \vdash}$ în aceleași condiții,

3′) $\dfrac{\alpha \vdash A \,;\, \alpha, A \vdash B}{\alpha \vdash B}$,

4′) $\dfrac{\alpha, A, B, \beta \vdash C}{\alpha, A \wedge B, \beta \vdash C}$,

5′) $\dfrac{\alpha \vdash A \wedge \sim A}{\alpha \vdash}$,

6′) $\dfrac{\alpha \vdash}{\alpha \vdash A}$,

7′) $\dfrac{\alpha, A \vdash}{\alpha \vdash \sim A}$,

8′) $\dfrac{\alpha \vdash A}{\alpha,\ \sim A \vdash}$,

9′) $\dfrac{\alpha, A \vdash B}{\alpha,\ \sim A \vdash\ \sim B}$,

10′) $\dfrac{\alpha,\ \sim A \vdash\ \sim B}{\alpha, A \vdash B}$.

Spre deosebire de regulile 1—12 introduse axiomatic, regulile admisibile se demonstrează cu ajutorul regulilor 1—12 inițiale. De exemplu, regula 3′) se demonstrează cu arborele

$$\dfrac{\dfrac{\dfrac{\alpha, A \vdash B}{\alpha \vdash A \to B}(7) \quad \alpha \vdash A}{\alpha \vdash B}(8)}{}$$

La demonstrarea regulii 6′) vom ține seama că secventul $\alpha \vdash$ fiind demonstrabil rezultă din $\alpha \vdash C$ și $\alpha \vdash\ \sim C$. Vom folosi deci ca secvenți inițiali acești secvenți demonstrabili, iar arborele următor va demonstra regula admisibilă căutată :

$$\dfrac{\dfrac{\dfrac{\alpha \vdash C}{\alpha,\ \sim A \vdash C}(12) \quad \dfrac{\alpha \vdash\ \sim C}{\alpha,\ \sim A \vdash\ \sim C}(12)}{\alpha,\ \sim A \vdash}(10)}{\alpha \vdash A}(9)$$

Demonstrarea regulii admisibile (7′) se face cu ajutorul următorului arbore, știind că $\alpha, A \vdash$ rezultă din $\alpha, A \vdash C$ și $\alpha, A \vdash\ \sim C$, conform regulii (10) :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha, A \vdash C}{\alpha, A,\ \sim\ \sim A \vdash C}(12)}{\alpha,\ \sim\ \sim A, A \vdash C}(11)}{\alpha,\ \sim\ \sim A \vdash A \to C}(7) \quad \dfrac{\alpha,\ \sim\ \sim A,\ \sim A \vdash}{\alpha,\ \sim\ \sim A \vdash A}(9)}{\alpha,\ \sim\ \sim A \vdash C}(8) \quad \dfrac{\dfrac{\dfrac{\alpha, A \vdash\ \sim C}{\alpha, A,\ \sim\ \sim A \vdash\ \sim C}}{\alpha,\ \sim\ \sim A,\ A \vdash\ \sim C} \quad \dfrac{\alpha,\ \sim\ \sim A,\ \sim A \vdash}{\alpha,\ \sim\ \sim A \vdash A}}{\alpha,\ \sim\ \sim A \vdash\ \sim C}}{\dfrac{\alpha,\ \sim\ \sim A \vdash}{\alpha \vdash\ \sim A}(9)}$$

Cu ajutorul regulilor admisibile sînt stabilite în continuare principalele propoziții ce caracterizează CP.

Introducînd substituția ca o aplicație $s : F \to F$, unde F este mulțimea formulelor, cu proprietățile de a păstra operațiile și de a se aplica la secvenți și arbori :

$$s(A \to B) = s(A) \to s(B), \quad s(A \vee B) = s(A) \vee s(B),$$

$$s(A \wedge B) = s(A) \wedge s(B), \quad s(\sim A) = \sim s(A),$$

$$s(A_1, \ldots, A_n \vdash A) = s(A_1), \ldots, s(A_n) \vdash s(A),$$

$$s(A_1, \ldots, A_n \vdash) = s(A_1), \ldots, s(A_n) \vdash,$$

Scanned with OKEN Scanner

$$s(\vdash A) = \vdash s(A),\ s(\vdash) = \vdash,$$

$$s\left(\frac{T_1, \ldots, T_k}{S}\right) = \frac{s(T_1) \ldots s(T_k)}{s(S)}$$

putem demonstra, folosind arborii,

Teorema substituției. *Dacă S este un secvent demonstrabil și s o substituție, secventul $s(S)$ este demonstrabil.*

Echivalența a două formule $A \equiv B$ este introdusă ca o proprietate de secvenți și anume prin cerința ca secvenții $A \vdash B$ și $B \vdash A$ să fie demonstrabili.

Se arată că, astfel definită, echivalența satisface proprietățile de reflexivitate, simetrie și tranzitivitate.

Se stabilește și o

Teoremă a înlocuirii. *Fie A o formulă și B o subformulă a sa, iar A' se obține din A prin înlocuirea unei apariții a lui B prin formula B'; dacă $B \equiv B'$, atunci $A \equiv A'$.*

Aplicații. a) Demonstrarea prin arbori de cvasiderivare a formulei $(A \to B) \equiv (\sim A \vee B)$:

$$\dfrac{\dfrac{\dfrac{A \vdash A;\ A \to B \vdash A \to B}{A \to B,\ A \vdash B}}{A \to B,\ A \vdash \sim A \vee B;} \qquad \dfrac{\dfrac{\sim A \vdash \sim A}{\sim A \vdash \sim A \vee B;}\quad \vdash A \vee \sim A}{A \to B \vdash \sim A \vee B};$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A;\ \sim A \vdash \sim A}{A,\ \sim A \vdash}}{A,\ \sim A \vdash B} \qquad B \vdash B;\ \sim A \vee B \vdash \sim A \vee B}{\sim A \vee B,\ A \vdash B}}{\sim A \vee B \vdash A \to B}$$

b) Demonstrarea prin arbori a distributivității:

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$

$$\dfrac{\dfrac{\dfrac{A \wedge (B \vee C) \vdash A;\ B \vdash B}{A \wedge (B \vee C),\ B \vdash A \wedge B} \qquad \dfrac{A \wedge (B \vee C) \vdash A;\ C \vdash C}{A \wedge (B \vee C),\ C \vdash A \wedge C}}{A \wedge (B \vee C),\ A \vdash (A \wedge B) \vee (A \wedge C);\ A \wedge (B \vee C),\ C \vdash (A \wedge B) \vee (A \wedge C);\ A \wedge (B \vee C) \vdash B \vee C};$$

$$\dfrac{\dfrac{\dfrac{A \wedge B \vdash B}{A \wedge B \vdash B \vee C}\ ;\ A \wedge B \vdash A \qquad \dfrac{A \wedge C \vdash C}{A \wedge C \vdash A \vee C}}{A \wedge B \vdash A \wedge (B \vee C);} \qquad \dfrac{A \wedge C \vdash C;}{A \wedge C \vdash A \wedge (B \vee C);\ (A \wedge B) \vee (A \wedge C) \vdash (A \wedge B) \vee (A \wedge C)}}{(A \wedge B) \vee (A \wedge C) \vdash A \wedge (B \vee C)}$$

2.10. TEORII FORMALIZATE

Fie $\mathscr{L}$ un limbaj formalizat definit de un alfabet și formulele sale $\mathscr{L} = \{A, F\}$.

Se numește *sistem deductiv bazat pe limbajul $\mathscr{L}$* perechea $\mathscr{S} = \{\mathscr{L}, C\}$, unde $\mathscr{L}$ este un limbaj, iar C o *operație de consecință*.

Scanned with OKEN Scanner

Numim *operație de consecință* o operație C care aplicată fiecărei mulțimi de formule H face să rezulte mulțimea $C(H)$ a tuturor consecințelor sale logice.

O operație de consecință satisface următoarele condiții:

1) $H \subset C(H)$,

2) dacă $H_1 \subset H_2$, atunci $C(H_1) \subset C(H_2)$,

3) dacă $H_1 \subset C(H_2)$ și $H_2 \subset C(H_3)$, atunci $H_1 \subset C(H_3)$:

Înlocuind în 3) $H_1 = C(C(H))$, $H_2 = C(H)$ și $H_3 = H$, obținem $C(C(H)) \subset C(H)$. Dar din 1) avem $C(H) \subset C(C(H))$ și deci

3′) $C(C(H)) = C(H)$.

Din 3′) cu ajutorul lui 2) putem obține 3). Avem $H_2 \subset C(H_3)$, $C(H_2) \subset C(C(H_3)) = C(H_3)$ și $C(H_1) \subset C(H_2) \subset C(H_3)$.

O altă proprietate este:

4) Dacă $A \subset C(H)$, există o submulțime finită $H_0 \subset H$ astfel ca $A \subset C(H_0)$. Aceasta este *proprietatea de finitudine a procesului de deducție*.

O operație de consecință cu aceste proprietăți este echivalentă cu noțiunea de *consecință logică*, dacă îndeplinește condiția:

a) pentru orice mulțime H de formule, $C(H)$ conține numai și numai acele formule A din limbaj, care sînt adevărate în toate interpretările limbajului în care toate formulele din H sînt adevărate.

Această condiție pentru cazul cînd H este vidă conduce la enunțul:

b) mulțimea $C(0)$ coincide cu mulțimea tautologiilor.

Completitudinea calculului propozițiilor ne îngăduie să modificăm definiția operației de consecință, ca rezultînd mulțimea tuturor formulelor derivabile din H prin regulile de inferență și axiomele sistemului. Această operație de consecință satisface cerințele 1), 2), 3), 3′) și 4).

Tripletul $\mathscr{T} = \{\mathscr{L}, C, \mathscr{A}\}$, unde $\mathscr{A}$ este o mulțime de tautologii privilegiate, numite *axiome*, este o *teorie formalizată*.

Dacă definim operația de consecință $C(H)$ ca producînd toate formulele derivabile din H prin regulile de inferență și din axiomele $\mathscr{A}$, atunci $C(H)$ este cea mai mică mulțime de formule conținînd pe H și $\mathscr{A}$, închisă față de fiecare din regulile de consecință.

Operația C poartă denumirea de logica teoriei $\mathscr{T}$, iar $C(\mathscr{A})$ este mulțimea teoremelor teoriei.

Cu aceste elemente, revedem întreprinderea acestui capitol în care a fost tratat calculul propozițiilor. Într-o definiție strictă acesta este sistemul deductiv $\mathscr{S} = \{\mathscr{L}, C\}$, unde $\mathscr{L}$ este un limbaj de ordinul 0. Celelalte sisteme logice ce vor fi considerate (în capitolele 6—10) sînt toate calcule propoziționale fiind sisteme deductive bazate pe limbaje de ordinul 0.

În același sistem deductiv, mulțimi diferite de axiome $\mathscr{A}$ și $\mathscr{A}'$ dau naștere la aceeași mulțime de teoreme $C(\mathscr{A}) = C(\mathscr{A}')$. Acesta este cazul mulțimilor de axiome echivalente din 2.3, care stau la baza calculului propozițional clasic.

Cînd $\mathscr{A} \subset \mathscr{A}'$, $C(\mathscr{A}) \subset C(\mathscr{A}')$ și deci toate teoremele teoriei $\mathscr{T} = \{\mathscr{L}, C, \mathscr{A}\}$ vor fi teoreme în $\mathscr{T}' = \{\mathscr{L}, C, \mathscr{A}'\}$. Aceasta este calea de a obține noi teorii din altele studiate, prin adăugarea de axiome suplimentare, ilustrate de legătura între diferitele sisteme logice modale descrise în cap. 8.

Teoriile matematice elementare (a mulțimilor, a aritmeticii, a ordonării, a laticelor, a grupurilor etc.) au toate forma $\{\mathscr{L}, C, \mathscr{A}\}$, dar limbajul lor nu este de ordinul 0, ci de ordinul 1. Ele se bazează totuși pe logica clasică (bivalentă), dar nu a propozițiilor, ci a predicatelor de ordinul 1.

Scanned with OKEN Scanner

Proprietăţile sistemului formal al calculului propoziţiilor au fost studiate pentru alte sisteme formale. Consistenţa, completitudinea, compactitatea se menţin şi în cazul sistemului formal al calculului cu predicate, scris pe un alfabet care cuprinde simboluri de cuantificatori, predicate, funcţii şi constante. Ceea ce se pierde la acest pas este decidabilitatea : nu mai există procedeu efectiv pentru a decide dacă o formulă dată este teoremă a sistemului. În continuare însă suprapunerea sintaxei pe semantică este perfectă : o teoremă este întotdeauna o formulă logic adevărată şi reciproc. În momentul cînd construim cu ajutorul sistemului calculului predicatelor un sistem formal pentru descrierea proprietăţilor numerelor naturale, un sistem pentru aritmetică, teorema de completitudine nu mai este valabilă, aşa cum a arătat Gödel (1930) : dacă sistemul este consistent este incomplet (există formule logic adevărate despre care nu putem spune dacă ele sau negaţia lor sînt teoreme). Următorul tablou rezumă această situaţie :

	Sistem formal Calculul propoziţiilor	Sistem formal Calculul predicatelor de ordinul 1	Sistem formal Calculul aritmetic
Consistenţă	+	+	+
Completitudine	+	+	−
Decidabilitate	+	−	−
Compactitate	+	+	+

Crearea sistemelor formale pentru calculul propoziţiilor şi calculul predicatelor cu proprietăţile de mai sus au produs un mare ecou. Toate ramurile ştiinţei au manifestat tendinţa precizării limbajului lor, construirii de sisteme formale cu axiome şi reguli de deducţie şi acceptarea drept teoreme sau teze numai a enunţurilor demonstrabile din axiome prin aplicarea regulilor.

2.11. NOTĂ ISTORICĂ

Primul sistem axiomatizat al calculului propoziţiilor se datorează lui G. Frege (1879), iar simplificarea sistemului pe care am folosit-o în acest capitol a fost făcută de J. Lukasiewicz în 1929. Tratatele de logică (A. Church, A. N. Prior) menţionează peste 30 sisteme echivalente mai importante elaborate pînă în 1965. Din ele am citat : Russell (1908), Hilbert-Ackermann (1928), Nicod (1918), Church (1932). Punctul de vedere sintactic al lui Frege a rămas mult timp fără ecou, dezvoltarea calculului propoziţiilor făcîndu-se în făgaşul semantic (E. Schröder, W. V. Quine etc.). Momentul de cristalizare poate fi considerat la apariţia lucrării *Principia Mathematica* (1910) de Russell şi Whitehead. Numele utilizat de Russell : „calculul propoziţional" (1903) a fost reţinut, deşi în varianta franceză s-a numit „calculul propoziţiilor " (Couturat).

La origine stă denumirea dată de Schröder : „Aussagenkalkul" (1890). Russell are multe ezitări în privinţa formulării regulii substituţiei, dar regula este reabilitată de C. I. Lewis (1918). Utilizarea schemelor de axiome

Scanned with OKEN Scanner

care fac regula substituției nenecesară a fost introdusă de J. von Neumann (1927) : procedeul de stabilire a independenței axiomelor a fost elaborat prima dată de P. Bernays (1926). Calculul bazat pe regulile de inferență, calculul natural a fost introdus de G. Gentzen și în paralel de J. Jaskowski.

Se utilizează pentru ceea ce am definit drept sistem formal denumirea de *sistem logistic*. La sfîrșitul secolului trecut au crescut preocupările pentru definirea celei mai mari precizii formale în matematică. Dată de cotitură poate fi considerat anul 1900, în jurul căruia David Hilbert și școala sa formalistă lansează programul, a cărui esență este căutarea de teorii care să aibă proprietățile formale de consistență și completitudine, proprietăți speciale menite să garanteze corectitudinea formală.

Teorema lui Gödel, a cărei consecință este că majoritatea teoriilor matematice nu sînt complete, a spulberat iluzia că programul lui Hilbert va putea fi aplicat. Teorema lui Gödel spune că pentru un sistem formal, ale cărui axiome definesc adecvat adunarea și înmulțirea numerelor naturale, există propoziții care sînt adevărate în $\langle \mathbb{N}, +, \cdot \rangle$, dar nu sînt demonstrabile din axiome prin aplicarea regulilor de inferență.

Cu toate acestea, interesul pentru axiomatizare și formalizare s-a manifestat puternic în toate domeniile științei, rămînînd și astăzi actuale. Conceptul de bază cu care lucrează în prezent filozofia științei și epistemologia este teoria științifică în sensul definiției date teoriei în cazul sistemelor formale.

Lui Hilbert i se datorează enunțarea clară a problemei deciziei (Entscheidungs Problem) ca problemă a găsirii unei proceduri (mecanice) efective pentru a determina validitatea (adevărul) unei formule în calculul predicatelor de ordinul întîi. Începînd cu teorema lui Gödel (între 1930—1940) multă atenție s-a acordat edificării unei teorii unitare a calculabilității, menite să definească teoretic ce este un „procedeu efectiv". Teoria recursivității, a algoritmilor, a limbajelor formale și diferitelor modele ale calculabilității, cum este mașina Turing, sînt achizițiile acestui efort. În toate aceste domenii ca și în matematică au fost identificate probleme nedecidabile.

Sistemul formal al calculului propozițilior a servit drept laborator pentru selectarea, enunțarea și testarea proprietăților importante, elucidate de A. Tarski, ale teoriilor formalizate. În afară de aceasta el a fost punctul de plecare al unor logici propoziționale construite prin părăsirea unor ipoteze fundamentale (de exemplu, bivalența în semantica calculului propozițiilor) și prin îmbogățirea, introducerea unor simboluri și axiome noi (de exemplu, logica modală).

Pentru inteligența artificială, cunoașterea sistemului formal al logicii propozițiilor este primul pas pentru familiarizarea cu problema fundamentală a modelării operațiilor inteligente pe calculator. Una din școlile influente în inteligența artificială pleacă de la presupunerea că mintea umană funcționează ca un sistem formal și poate fi modelată în mod corespunzător.

2.12. EXERCIȚII ȘI PROBLEME

1. Să se demonstreze că

$$\sim B \to (B \to A) \text{ și } (\sim A \to A) \to A$$

sînt teoreme în **L**.

Scanned with OKEN Scanner

2. Să se folosească teorema deducției pentru a arăta că următoarele formule sînt teoreme în **L**:

a) $((A \to B) \to A) \to A$,

b) $\sim(\sim A \to B) \to (B \to A)$.

3. Să se efectueze demonstrațiile pentru următoarele formule:

a) $(p \to q) \to ((\sim p \to \sim q) \to (q \to p))$,

b) $((p \to (q \to r)) \to (p \to q)) \to ((p \to (q \to r)) \to (p \to r))$.

4. Să se arate că

a) $\{A \to B,\ \sim(B \to C) \to \sim A\} \underset{\mathbf{L}}{\vdash} (A \to C)$,

b) $\{A \to (B \to C)\} \underset{\mathbf{L}}{\vdash} (B \to (A \to C))$.

5. a) Să se arate că sistemul Hilbert-Ackermann are aceleași teoreme ca și **L**.

b) Să se arate că sistemul (Rosser), avînd conectori primitivi $\to$, $\wedge$, $\vee$ și $\sim$, regula MP și axiomele:

1) $A \to (B \to A)$,

2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$,

3) $A \wedge B \to A$,

4) $A \wedge B \to B$,

5) $A \to (B \to (A \wedge B))$,

6) $A \to A \vee B$.,

7) $B \to A \vee B$,

8) $(A \to C) \to ((B \to C) \to (A \vee B \to C))$,

9) $((A \to B) \to ((A \to \sim B) \to \sim A))$,

10) $\sim\sim A \to A$,

are aceleași teoreme ca sistemul **L**.

6. Să se arate că sistemul (Rasiowa, Sikorski), avînd conectori primitivi $\to$, $\wedge$, $\vee$ și $\sim$, regula MP și axiomele:

1) $(A \to B) \to ((B \to C) \to (A \to C))$,

2) $A \to A \vee B$,

3) $B \to A \vee B$,

4) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$,

5) $A \wedge B \to A$,

6) $A \wedge B \to B$,

7) $(C \to A) \to ((C \to B) \to (C \to (A \wedge B)))$,

8) $(A \to (B \to C)) \to ((A \wedge B) \to C)$,

9) $((A \wedge B) \to C) \to (A \to (B \to C))$,

10) $(A \wedge \sim A) \to B$,

11) $(A \to (A \wedge \sim A)) \to \sim A$.

12) $A \vee \sim A$.

are aceleași teoreme ca sistemul **L**.

7. Să se arate că axioma L3 a sistemului **L** poate fi înlocuită de axioma

$$\text{L3}'\ (\sim B \to \sim A) \to ((\sim B \to A) \to B)$$

fără ca și clasa teoremelor să se schimbe.

Scanned with OKEN Scanner

8. Să se demonstreze echivalența lui **L** cu sistemul cu aceiași conectori, cu regulile de inferență MP și substituție și cu axiomele :

1) $p \to (q \to p)$,

2) $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$,

3) $(\sim q \to \sim p) \to ((\sim q \to p) \to q)$.

9. Să se demonstreze că mulțimea teoremelor sistemului $\mathbf{L}^+$, obținut prin includerea formulei A din **L** ca axiomă, este diferită de mulțimea teoremelor din **L** numai și numai dacă A nu este teoremă în **L**.

10. Să se extindă pe **L** prin adăugarea lui $A = (\sim p \to q) \to (p \to \sim q)$ ca axiomă. Să se caracterizeze noul sistem $\mathbf{L}^+$ în raport cu **L**.

11. Să se arate că :

a) $\sim \sim A \equiv A$,

b) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$,

c) $\sim (A \vee B) \equiv \sim A \wedge \sim B$

prin metoda arborilor.

BIBLIOGRAFIE

1 Church A. *Introduction to Mathematical Logic*. Princeton, Univ. Press, 1956.
2. Enderton, H. *A Mathematical Introduction to Logic*. New York, Academic Press, 1972.
3. Ersov, Y., Paliutin, E. *Mathematical Logic*. Moscow, Mir, 1984.
4. Grzegorczyk, A. *An Outline of Mathematical Logic*. Warszawa, Polish Sc. Publisher, 1974.
5. Hamilton, A. G. *Logic for Mathematicean*. London, Cambridge University Press, 1978.
6. Kleene, S. C. *Mathematical Logic*. New York, John Wiley, 1967.
7. Mendelson, E. *Introduction to Mathematical Logic*. New York, van Nostrand, 1965.
8. Quine, W. O. *Mathematical Logic*. Cambridge Harward, University Press, 1958.
9. Tarski, A. *Introduction to Logic*. New York, Oxford University Press, 1976.
10. Tîrnoveanu, M. *Elemente de logică matematică*, vol. I : *Logica propozițiilor bivalente*. București, Editura Didactică și Pedagogică, 1964.

Scanned with OKEN Scanner

3.

LATICE ȘI ALGEBRE BOOLEENE

3.1. LATICE ORDONATE ȘI LATICE ALGEBRICE

Se numește *latice algebrică* $(L, \cap, \cup)$ o mulțime nevidă L cu două operații $\cap$ (produs) și $\cup$ (sumă) cu următoarele proprietăți pentru toți $x, y, z \in L$:

comutativitate : $x \cap y = y \cap x, \quad x \cup y = y \cup x,$

asociativitate : $x \cap (y \cap z) = (x \cap y) \cap z,$

$x \cup (y \cup x) = (x \cup y) \cup z,$

absorbție : $x \cap (x \cup y) = x, \; x \cup (x \cap y) = x,$

idempotență $x \cap x = x, \quad x \cup x = x.$

Se numește *latice ordonată* $(L, \leqslant)$ o mulțime cu o relație de ordine parțială, $\leqslant$, în care pentru fiecare pereche de elemente x, y din L există $\sup(x, y)$ și $\inf(x, y)$.

O relație R pe o mulțime A este *o relație de ordine parțială* dacă R este reflexivă (pentru toți $a \in A$ avem aRa), antisimetrică (pentru toți $a \in A$, aRb și bRa implică $a = b$) și tranzitivă (pentru toți $a, b, c \in A$, aRb și bRc implică aRc).

Într-o mulțime ordonată A, elementul $a \in A$ este *cel mai mare element* al lui A, dacă pentru toți $x \in A$ avem $x \leqslant a$. Elementul $b \in A$ este *cel mai mic element* al lui A, dacă pentru toți $x \in B$ avem $x \geqslant b$. Un element c este *maximal*, cînd $c \leqslant x$ implică $c = x$ pentru toți $x \in A$; un element d este *minimal* cînd $x \leqslant d$ implică $x = d$ pentru toți $x \in A$. Fiecare cel mai mare element este maximal și fiecare cel mai mic element este minimal. În timp ce mulțimea ordonată are cel puțin un cel mai mare și un cel mai mic elemente, poate să nu aibă elemente maximale și minimale sau să aibă mai multe.

Supremum (notat sup B) al unei submulțimi $B \subseteq A$ este cel mai mic element al marginilor superioare ale lui B definite astfel: $a \in A$ este o margine superioară a lui B dacă pentru toți $b \in B$ avem $a \geqslant b$.

Infimum (notat inf B) al unei mulțimi $B \subseteq A$ este cel mai mare element al marginilor inferioare ale lui B definite astfel: $a \in A$ este o margine inferioară a lui B dacă pentru toți $b \in B$ avem $a \leqslant b$.

Într-o latice ordonată sînt echivalente propozițiile:

a) $x \leqslant y$; b) $\sup(x, y) = y$; c) $\inf(x, y) = x$.

Scanned with OKEN Scanner

O relație de ordine parțială $\leqslant$ in A este numită *ordine totală*, dacă pentru orice $a, b \in A$ avem $a \leqslant b$ sau $b \leqslant a$. $(A, \leqslant)$ se numește *mulțime total ordonată* sau *lanț*. Unii autori numesc mulțimea cu o relație de ordine totală mulțime ordonată și mulțimea cu o relație de ordine parțială mulțime parțial ordonată.

O relație binară $\leqslant$ care este reflexivă și tranzitivă se numește relație de *cvasiordine* sau preordine, iar mulțimea $(A, \leqslant)$ cu o relație de cvasiordine este *cvasiordonată*.

Lema lui Zorn sau principiul inducției pentru mulțimile ordonate. *Dacă fiecare mulțime total ordonată (lanț) de elemente a unei mulțimi ordonate A (cu relație de ordine parțială) are o margine superioară în A, atunci mulțimea A conține cel puțin un element maximal. Pentru fiecare $a_0 \in A$ există un element maximal $a \geqslant a_0$.*

Teoremă. *Fie $(L, \leqslant)$ o latice ordonată. Dacă definim*

$$x \cap y = \inf(x, y) \text{ și } x \cup y = \sup(x, y),$$

laticea $(L, \cap, \cup)$ este o latice algebrică.

Prin definițiile de mai sus se introduc două operații care satisfac proprietățile cerute de operațiile laticei algebrice.

Comutativitatea:

$$x \cap y = \inf(x, y) = \inf(y, x) = y \cap x$$

$$x \cup y = \sup(x, y) = \sup(y, x) = y \cup x, \qquad \forall x, y \in L.$$

Asociativitatea:

$$x \cap (y \cap z) = x \cap \inf(y, z) = \inf(x, \inf(y, z)) = \inf(x, y, z) =$$

$$= \inf(\inf(x, y), z) = \inf(x, y) \cap z = (x \cap y) \cap z;$$

$$x \cup (y \cup z) = x \cup \sup(y, z) = \sup(x, \sup(y, z)) = \sup(x, y, z) =$$

$$= \sup(\sup(x, y), z) = \sup(x, y) \cup z = (x \cup y) \cup z, \qquad \forall x, y, z \in L.$$

Absorbția:

$$x \cap (x \cup y) = x \cap \sup(x, y) = \inf(x, \sup(x, y)) = x;$$

$$x \cup (x \cap y) = x \cup \inf(x, y) = \sup(x, \inf(x, y)) = x, \quad \forall x, y \in L.$$

Idempotența:

$$x \cap x = \inf(x, x) = x, \qquad x \cup x = \sup(x, x) = x, \qquad \forall x \in L.$$

Teoremă. *Fie $(L, \cap, \cup)$ o latice algebrică. Dacă definim relația de ordine parțială $x \leqslant y$ prin*

$$x \cap y = x \text{ sau } x \cup y = y,$$

atunci $(L, \leqslant)$ este o latice ordonată.

Verificăm dacă relația $\leqslant$ satisface proprietățile de ordine parțială.

Reflexivitatea: $x \cap x = x$ și $x \cup x = x$, deci $x \leqslant x$.

Antisimetria: $x \leqslant y$ și $y \leqslant x$ înseamnă că $x \cap y = x$ și $y \cap x = y$, dar $x \cap y = y \cap x$ și deci $x = y$.

Tranzitivitatea: $x \leqslant y$ și $y \leqslant z$ sînt scrise drept $x \cap y = x$ și $y \cap z = y$. Deci $x = x \cap y = x \cap (y \cap z) = (x \cap y) \cap z = x \cap z$ sau $x \leqslant z$.

Scanned with OKEN Scanner

Existența $\sup(x, y)$ și $\inf(x, y)$ rezultă din :

$x \cap (x \cup y) = x$ implică $x \leqslant x \cup y$,

$y \cap (x \cup y) = y$ implică $y \leqslant x \cup y$.

Fie $z \in L$ cu proprietatea $x \leqslant z$ și $y \leqslant z$. Din

$$(x \cup y) \cup z = x \cup (y \cup z) = x \cup z = z$$

deducem $x \cup y \leqslant z$, deci $\sup(x, y) = x \cup y$. La fel $\inf(x, y) = x \cap y$.

Cele două teoreme îngăduie ca de acum înainte să vorbim de latice, fără să specificăm dacă laticea este construită printr-o relație de ordine parțială sau prin cele două operații binare.

Diagrama Hasse permite o reprezentare grafică a unei mulțimi ordonate. Elementele mulțimii sînt reprezentate drept puncte, iar relația $a < b$ ($a \leqslant b$, $a \neq b$) printr-o linie care leagă cele două puncte, b fiind situat mai sus ca a. Diagramele Hasse (fig. 3.1) sînt în special utile pentru reprezentarea laticelor și recunoașterea proprietăților lor.

În fig. 3.1 mulțimile parțial ordonate nu sînt latice. În fig. 3.1, a elementul a este minimal și cel mai mic, b și d sînt maximale, dar nu există cel mai mare element. În fig. 3.1, b elementele a și b sînt maximale, e și f minimale, dar nu există cel mai mare sau cel mai mic element. În fig. 3.1, c condiția de latice nu există, neexistînd $\sup(a, b)$.

Fig. 1

Definiție. Dacă o latice conține un cel mai mic element în raport cu $\leqslant$, acesta este numit *element zero* și este notat cu 0.

Dacă există cel mai mare element în raport cu $\leqslant$, acesta este numit *element unitate* și este notat cu 1.

Orice latice finită L are un 0 și un 1. Toate elementele sale, $x \in L$, satisfac relațiile $x \leqslant 1$, $0 \cap x = 0$, $0 \cup x = x$, $1 \cap x = x$ și $1 \cup x = 1$.

Exemple. Fie laticele cu 5 elemente (fig. 3.2). Laticea V_3^5 se numește *latice diamant*, V_4^5 *latice pentagon* iar V_5^5 este un lanț. Tabla de operații pentru matricea pentagon V_4^5 este următoarea :

$\cap$	0	a	b	c	1
0	0	0	0	0	0
a	0	a	0	c	a
b	0	0	b	0	b
c	0	c	0	c	c
1	0	a	b	c	1

$\cup$	0	a	b	c	1
0	0	a	b	c	1
a	a	a	1	a	1
b	b	1	b	1	1
c	c	a	1	c	1
1	1	1	1	1	1

Cele două matrice permit să citim $\inf(x, y) = x \cap y$ și $\sup(x, y) = x \cup y$ pentru fiecare din perechile (x, y). De exemplu $\inf(a, b) = 0$ și $\sup(a, b) = 1$, ceea ce rezultă și din diagramă.

Scanned with OKEN Scanner

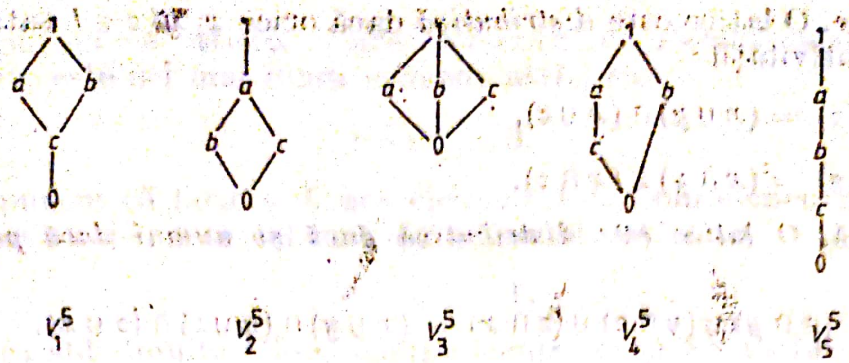


Fig. 3.2

Principiul dualității enunțat pentru mulțimile ordonate este următorul :

Orice enunț cu privire la $(A, \leqslant)$ *rămîne corect dacă în cuprinsul său înlocuim relația* $\leqslant$ *cu relația inversă* $\geqslant$. Mulțimea $(A, \geqslant)$ este numită *duala* lui $(A, \leqslant)$.

Enunțat pentru latice, principiul dualității devine :

Orice enunț (formulă, proprietate) cu privire la o latice $(L, \cap, \cup)$ *rămîne valabil dacă peste tot în cuprinsul său schimbăm între ele simbolurile* $\cap$ *și* $\cup$.

3.2. LATICE MODULARE, DISTRIBUTIVE, COMPLEMENTATE

Teoremă. *În orice latice sînt satisfăcute inegalitățile distributivității :*

$$x \cap (y \cup z) \geqslant (x \cap y) \cup (x \cap z), \quad x \cup (y \cap z) \leqslant (x \cup y) \cap (x \cup z).$$

Demonstrăm prima din inegalități. Elementul $x \cap (y \cup z)$ este o margine superioară pentru $x \cap y$ și $x \cap z$. Într-adevăr :

$$x \cap y \leqslant x, \quad x \cap y \leqslant y \leqslant y \cup z, \quad \text{deci } x \cap y \leqslant x \cap (y \cup z),$$

$$x \cap z \leqslant x \leqslant x \cap z \leqslant z \leqslant y \cup z, \quad \text{deci } x \cap z \leqslant x \cap (y \cup z).$$

Rezultă $x \cap (y \cup z) \geqslant (x \cap y) \cup (x \cap z)$.

Corolar. *În orice latice sînt satisfăcute inegalitățile modularității :*

dacă $x \geqslant z$, *atunci* $x \cap (y \cup z) \geqslant (x \cap y) \cup z$,

dacă $x \leqslant z$, *atunci* $x \cup (y \cap z) \leqslant (x \cup y) \cap z$.

Rezultă din teoremă, ținînd seama că

dacă $x \geqslant z$, atunci $(x \cap y) \cup (x \cap z) = (x \cap y) \cup z$,

dacă $x \leqslant z$, atunci $(x \cup y) \cap (x \cup z) = (x \cup y) \cap z$.

Definiție. O latice este *modulară*, dacă toți $x, y, z \in L$, cînd $x \leqslant z$, satisfac *ecuația modularității* $x \cup (y \cap z) = (x \cup y) \cap z$.

Teoremă. *O latice este modulară dacă pentru toți* $x, y, z \in L$ *avem*

$$x \cup (y \cap (x \cup z)) = (x \cup y) \cap (x \cup z).$$

Înlocuim în ecuația modularității pe z cu $x \cup z$, condiția $x \leqslant x \cup z$ fiind îndeplinită.

Scanned with OKEN Scanner

Definiție. O latice este *distributivă* dacă orice $x, y, z \in L$ satisfac ecuațiile distributivității :

$$x \cup (y \cap z) = (x \cup y) \cap (x \cup z),$$

$$x \cap (y \cup z) = (x \cap y) \cup (x \cap z).$$

Teoremă. *O latice este distributivă dacă și numai dacă pentru orice* $x, y, z \in L$

$$(x \cap y) \cup (y \cap z) \cup (z \cap x) = (x \cup y) \cap (y \cup z) \cap (z \cup x).$$

Corolar. Orice latice distributivă este modulară.

Teoremă. *O latice este distributivă dacă și numai dacă este satisfăcută regula reducerii* :

$$din\ x \cap y = x \cap z\ și\ x \cup y = x \cup z\ rezultă\ y = z.$$

Exemple. 1) Laticea pentagon nu este modulară. Pentru $c \leqslant a$

$$c \cup (b \cap a) = c \cup 0 = c \quad \text{și} \quad (c \cup b) \cap a = 1 \cap a = a.$$

Această proprietate este luată drept criteriu pentru recunoașterea oricărei latice nemodulare, care are o sublatice izomorfă cu laticea pentagon.

2) Laticea tuturor subspațiilor unui spațiu vectorial este modulară.

3) Laticea tuturor subgrupurilor unui grup este în general nemodulară.

4) Laticea submulțimilor unei mulțimi $(\mathcal{P}(M), \cap, \cup)$ este o latice distributivă (și deci modulară, v. fig. 3.3).

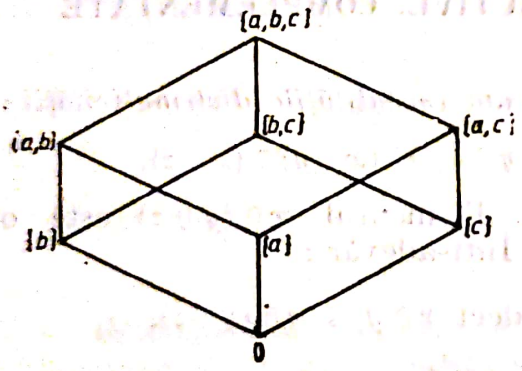


Fig. 3

5) Laticea diamant V_3^5 nu este distributivă (deși modulară). Într-adevăr, $a \cup (b \cap c) = a \cup 0 = a$, iar $(a \cup b) \cap (a \cup c) = 1 \cap 1 = 1$. Proprietatea este luată drept criteriu de recunoaștere a laticelor distributive : o latice modulară este distributivă dacă și numai dacă nici una din sublaticele ei nu este izomorfă cu laticea diamant.

Definiție. O latice L (cu 0 și 1) este *complementată* dacă pentru fiecare $x \in L$ există cel puțin un element y, numit *complementul lui* x, astfel ca $x \cap y = 0$, $x \cup y = 1$.

Teoremă. *O latice distributivă are pentru orice* $x \in L$ *cel mult un complement.*

Dacă ar avea doi complementi y_1, y_2, atunci $x \cup y_1 = x \cup y_2 = 1$ și $x \cap y_1 = x \cap y_2 = 0$. Astfel, conform regulii reducerii $y_1 = y_2$. Distributivitatea asigură deci existența complementului unic al lui x, notat x'.

3.3. LATICE RELATIV PSEUDOCOMPLEMENTATE ȘI ALGEBRE PSEUDOBOOLEENE

Introducerea noțiunii de *mulțime complementată* în teoria mulțimilor se face prin două definiții care coincid : a) complementara unei mulțimi A față de U este cea mai mare mulțime disjunctă A', $A' \subset U$, cu $A \cap A' = \emptyset$, sau b) cea mai mică submulțime A' a lui U, cu $A \cup A' = U$.

În cazul laticelor definițiile analoage cu cele de mai înainte conduc la două tipuri de complemente : unul față de produs și altul față de sumă.

Scanned with OKEN Scanner

Presupunem că laticea L are elementul 0. *Pseudocomplementul lui* x, notat $x^{\cap}$, este cel mai mare element astfel ca

$$x \cap x^{\cap} = 0.$$

Presupunem că laticea L are elementul 1. Complementul lui x față de $\cup$, notat cu $x^{\cup}$, este cel mai mic element astfel ca

$$x \cup x^{\cup} = 1.$$

Dacă există simultan două complemente $x^{\cap}$ și $x^{\cup}$ ale lui x, în general acestea nu sînt egale: $x^{\cap} \neq x^{\cup}$.

Dacă laticea L este distributivă, avem $x^{\cap} \leqslant x^{\cup}$ pentru că

$x^{\cap} = x^{\cap} \cap 1 = x^{\cap} \cap (x \cup x^{\cup}) = (x^{\cap} \cap x) \cup (x^{\cap} \cup x^{\cup}) = 0 \cup (x^{\cap} \cap x^{\cup}) = x^{\cap} \cup x^{\cup}$ și deci $x^{\cap} \leqslant x^{\cup}$.

Cînd cele două complemente sînt egale, avem $x^{\cap} = x^{\cup} = x'$ cu proprietățile ce definesc în mod unic complementul

$$x \cap x' = 0, \quad x \cup x' = 1.$$

Denumirea de pseudocomplement este rezervată lui $x^{\cap}$, care joacă un rol important în definirea unor latice și algebre speciale. Definirea prin care a fost introdus are ca urmare proprietatea

$$u \leqslant x^{\cap} \text{ dacă și numai dacă } u \cap x = 0$$

și, în particular, $x \cap x^{\cap} = 0$.

Teoremă. *Într-o latice în care* x, y *au pseudocomplementele* $x^{\cap}$ *și* $y^{\cap}$, $x \leqslant y$ *implică* $y^{\cap} \leqslant x^{\cap}$.

Într-adevăr, $x \cap y^{\cap} = (x \cap y) \cap y^{\cap} = x \cap (y \cap y^{\cap}) = x \cap 0 = 0$. Dar conform proprietății enunțate, $y^{\cap} \leqslant x^{\cap}$.

Proprietăți ale complementului. Dacă x, y au complementele x' și y' într-o latice distributivă, atunci:

1) $x \leqslant y$ dacă și numai dacă $y' \leqslant x'$,

2) $x'' = x$,

3) $(x \cup y)' = x' \cap y'$ și $(x \cap y)' = x' \cup y'$.

Proprietatea 2) rezultă din $x \cap x' = x' \cap x = 0$ și $x \cup x' = x' \cup x = 1$, deci x este complementul lui x', adică $(x')' = x$.

Pentru 1) trebuie să arătăm doar că $y' \leqslant x'$ implică $x \leqslant y$ pentru că reciproca fiind valabilă pentru pseudocomplemente este valabilă și pentru complemente. Dar $y' \leqslant x'$ implică $x'' \leqslant y''$ și, conform cu 2), $x \leqslant y$.

Să arătăm că complementul sumei 3) este $x' \cap y'$:

$$(x \cup y) \cap (x' \cap y') = (x \cap x' \cap y') \cup (y \cap x' \cap y') = 0,$$

$$(x \cup y) \cup (x' \cap y') = (x \cup y \cup x') \cap (x \cup y \cup y') = 1.$$

La fel se arată pentru complementul produsului.

Pseudocomplement relativ. Prin analogie cu definiția pseudocomplementului (față de 0) se introduce pseudocomplementul unui element față pe un alt element y, ca cel mai mare element z astfel ca $x \cap z \leqslant y$. Elementul z se mai numește *pseudocomplementul lui* x *modulo* y și se notează

$$z = x \to y.$$

Scanned with OKEN Scanner

Pentru orice $u \in L$,

$$u \leqslant x \to y \text{ dacă și numai dacă } x \cap u \leqslant y,$$

este o altă formă a definiției pseudocomplementului relativ la y.

O dată ce $x \cap y \leqslant y$, rezultă că $y \leqslant x \to y$, cînd acesta există.

Dacă laticea L are elementul 1, atunci:

a) $x \to x = 1$,

b) $x \leqslant y$ dacă și numai dacă $x \to y = 1$

c) $1 \to y = y$.

Dacă laticea L are elementul 0, atunci

$$x^{\cap} = x \to 0,$$

ceea ce confirmă că pseudocomplementul relativ este o generalizare a pseudocomplementului.

Elementul $x \to y$ nu există întotdeauna. De exemplu, dacă L nu are 1, elementul $x \to x$ nu există.

Latice relativ pseudocomplementată. O latice L este *relativ pseudocomplementată* dacă, oricare ar fi $x, y \in L$, există $x \to y$. Orice latice pseudocomplementată are elementul 1.

Orice latice relativ pseudocomplementată este distributivă:

$$x \cap (y \cup z) = (x \cap y) \cup (x \cap z).$$

Într-adevăr elementul $x \to u$, unde $u = (x \cap y) \cup (x \cap z)$, există. Atunci $x \cap y \leqslant u$ și $x \cap z \leqslant u$, $y \leqslant x \to u$ și $z \leqslant x \to u$.

Deci $y \cup z \leqslant x \to u$ și $x \cap (y \cup z) \leqslant u$ conform definiției lui $x \to u$. Pe de altă parte

$$x \cap y \leqslant x \cap (y \cup z), \quad x \cap z \leqslant x \cap (y \cup z)$$

și $u \leqslant x \cap (y \cup z)$, ceea ce demonstrează distributivitatea lui L, pentru că x, y, z sînt elemente oarecare ale lui L.

Într-o latice relativ pseudocomplementată se demonstrează următoarele proprietăți:

1) $x \to y = 1$ dacă și numai dacă $x \leqslant y$.

2) $x = y$ dacă și numai dacă $x \to y = y \to x = 1$.

3) $x \to x = 1$.

4) $x \to 1 = 1$.

5) $1 \to y = y$.

6) $(x \to x) \cap y = y$.

7) $x \cap (x \to y) \leqslant y$.

8) Dacă $x_1 \leqslant x_2$, atunci $x_2 \to y \leqslant x_1 \to y$.

9) Dacă $y_1 \leqslant y_2$, atunci $x \to y_1 \leqslant x \to y_2$.

10) $y \leqslant x \to y$.

11) $x \cap (x \to y) = x \cap y$.

Scanned with OKEN Scanner

12) $(x \to y) \cap y = y$.

13) $(x \to y) \cap (x \to z) = x \to (y \cap z)$.

14) $(x \to z) \cap (y \to z) = (x \cup y) \to z$.

15) $x \to (y \to z) = (x \cap y) \to z = y \to (x \to z)$.

16) $z \to x \leqslant (z \to (x \to y)) \to (z \to y)$.

17) $(x \to y) \cap (y \to z) \leqslant x \to z$.

18) $(x \to y) \leqslant (y \to z) \to (x \to z)$.

19) $x \leqslant y \to (x \cap y)$.

20) $x \to (y \to z) \leqslant (x \to y) \to (x \to z)$.

21) $z \cap ((z \cap x) \to (z \cap y)) = z \cap (x \to y)$.

Vom demonstra cîteva din aceste proprietăţi.

8). Din 7) rezultă $x_2 \cap (x_2 \to y) \to y$ şi dacă $x_1 \leqslant x_2$, atunci $x_1 \cap (x_2 \to y) \leqslant y$ sau, conform definiţiei, $x_2 \to y \leqslant x_1 \to y$. La fel se demonstrează 9).

13). Din 9) şi $y \cap z \leqslant y$, $y \cap z \leqslant z$ rezultă $x \to (y \cap z) \leqslant x \to y$ şi $x \to (y \cap z) \leqslant x \to z$. Deci $x \to (y \cap z) \leqslant u$, unde $u = (x \to y) \cap (x \to z)$, iar $u \leqslant x \to y$ şi $u \leqslant x \to z$. De aici rezultă $x \cap u \leqslant y$, $x \cap u \leqslant z$ şi $x \cap u \leqslant y \cap z$ sau $u \leqslant x \to (y \cap z)$, ceea ce arată că $u = x \to (y \cap z)$.

16). Folosim 9) : $z \to (x \cap y) \leqslant z \to y$, 11) : $z \to (x \cap (x \to y)) \leqslant z \to y$ şi 13) : $(z \to x) \cap (z \to (x \to y)) \leqslant z \to y$.

20). Din 11) : $x \cap (x \to y) \cap (x \to (y \to z)) = x \cap y \cap (y \to z) = x \cap y \cap z \leqslant z$ şi, conform definiţiei, $(x \to y) \cap (x \to (y \to z)) \leqslant x \to z$, deci $x \to (y \to z) \leqslant (x \to y) \to (x \to z)$.

Algebre pseudobooleene. O latice relativ pseudocomplementată care are pe lîngă elementul 1 şi elementul 0 se numeşte *algebră pseudobooleană*. Pseudocomplementul x^0 există pentru orice $x \in L$, fiind

$$x^0 = x \to 0.$$

Toate proprietăţile demonstrate pentru laticele relativ pseudocomplementate sînt valabile şi pentru algebrele pseudobooleene. În plus, se pot demonstra următoarele proprietăţi :

1') Dacă $x \leqslant y$, atunci $y^0 \leqslant x^0$.

2') $0^0 = 1$ şi $1^0 = 0$.

3') $x \cap x^0 = 0$.

4') $(x \cap x^0)^0 = 1$.

5') $x \leqslant x^{0\,0}$.

6') $x^{0\,0\,0} = x^0$.

7') $(x \cup y)^0 = x^0 \cap y^0$.

8') $x^0 \cup y^0 \leqslant (x \cap y)^0$.

9') $x^0 \cup y \leqslant x \to y$.

10') $x \to y \leqslant y^0 \to x^0$.

11') $x \to y^0 \leqslant (x \cap y)^0 = y \to x^0$.

12') $x \to y^0 = (x \to y^0)^{0\,0}$.

13') $(x \to y)^{0\,0} \leqslant x \to y^{0\,0}$.

14') $0 \to x = 1$.

Vom demonstra cîteva din aceste proprietăți.

5′). Deoarece $x \cap x^{\cap} \leqslant 0$ (3′), urmează $x \leqslant x^{\cap} \to 0 = x^{\cap\cap}$ prin definiția lui $x^{\cap}$ (în cazul complementului $x'' = x$).

8′). $(x^{\cap} \cup y^{\cup}) \cap (x \cap y) \leqslant 0$ prin distributivitate și 3′). Rezultă $x^{\cap} \cup y^{\cap} \leqslant (x \cap y) \to 0 = (x \cap y)^{\cap}$ (în cazul complementului $x' \cup y' = (x \cap y)'$).

10′). Rezultă din 7), unde $z = 0$, $(x \to y) \cap y^{\cap} \leqslant x^{\cap}$ și, conform definiției, $(x \to y) \leqslant y^{\cap} \to x^{\cap}$ (în cazul complementului $x \to y = y' \to x'$).

13′). $x \to y \leqslant x \to y^{\cap\cap}$ din 5′) și 9′) Prin 1′) și 12′) : $(x \to y)^{\cap\cap} \leqslant (x \to y^{\cap\cap})^{\cap\cap} = x \to y^{\cap\cap}$ (în cazul complementului desigur $x \to y = x \to y''$).

Comentariu. Paralelismul între operatorii unei latice $\cap$ și $\cup$ și conectorii unei logici $\wedge$ și $\vee$, scos în evidență de proprietățile de comutativitate, asociativitate, absorbție, idempotență (v. 3.1), a fost completat în cazul distributivității existente întotdeauna la conectorii logici și în acele latice care sînt distributive.

Proprietatea echivalenței relației $x \to y = 1$ cu $x \leqslant y$ a oricărei latice, care are elementul 1, ne dezvăluie corespondența care există între relația de ordine parțială din latice și relația de consecință logică în calculul propozițiilor, unde $x \Rightarrow y$ a fost definit ca o implicație $x \to y$ tautologică.

Faptul că negația din logica clasică este analogul complementului din laticele distributive complementate mărește interesul pentru cele din urmă.

3.4. ALGEBRE BOOLEENE

Definiție. O *algebră booleană* $(B, \cap, \cup, 0, 1, ')$ este o latice distributivă complementată. Proprietățile laticelor distributive, modulare, complementate sînt valabile în algebrele booleene. Astfel complementul fiecărui element este unic.

O latice relativ pseudocomplementată poate fi concepută ca o algebră $(L, \cap, \cup, \to, 1)$ cu trei operații binare, iar o algebră pseudobooleană ca o algebră $(L, \cap, \cup, \to, 1, 0, ^{\cap})$ cu trei operații binare și o operație unară : pseudocomplementul$^{\cap}$.

Orice algebră booleană B este o algebră pseudobooleană, unde $x \to y = x' \cup y$ și $x' = x \to 0$.

Dacă L este o algebră pseudobooleană și pentru orice $x \in L$ pseudocomplementul $x^{\cap}$ este complementul lui x, x', atunci L este o algebră booleană.

Toate proprietățile valabile pentru laticele relativ pseudocomplementate și pentru algebrele pseudoboolene sînt valabile pentru algebrele booleene.

În oricare algebră booleană avem în plus :

1″) $x \leqslant y$ dacă și numai dacă $y' \leqslant x'$.

2″) $x \cup x' = 1$.

3″) $x'' = x$.

4″) $x \to y = y' \to x'$.

5″) $x' \to y = y' \to x$.

6″) $(x \cap y)' = x' \cup y'$.

7″) $x \cup y = (x' \cap y')'$.

8″) $x \cap y = (x' \cup y')'$.

9″) $(x \to y)' = x \cap y'$.

10″) $(x \to y) \to x = x$.

Scanned with OKEN Scanner

Vom demonstra cîteva din aceste proprietăți :

4''). Rezultă din identitatea $x \to y' = y \to x'$, care a fost demonstrată pentru algebrele pseudobooleene 11'). Înlocuim pe y cu y' și folosim 3'') : $x \to y'' = y' \to x'$ și $x \to y = y' \to x'$. Identitățile 4) și 5) se numesc *legile contrapunerii*.

8''). Se știe că $(x \cup y)' = x' \cap y'$ din proprietățile algebrelor pseudobooleene 7'). Înlocuim pe x cu x' și pe y cu y' și aplicăm 3''). Obținem $(x' \cup \cup y')' = x'' \cap y'' = x \cap y$. Identitățile 6''), 7'') și 8'') se numesc *legile lui De Morgan*.

În afara modului de a introduce algebrele booleene ca latice distributive și complementate, există posibilitatea de a le defini prin următoarele cinci axiome : $\langle B, \cap, \cup, ' \rangle$ este o algebră booleană, unde B este o mulțime de elemente, iar $\cap, \cup, '$ operatori binari și unari, care satisfac următoarele axiome :

1° $\cap$ este o operație comutativă.

2° $\cap$ este o operație asociativă.

3° Pentru toți $x, y \in B$, dacă $x \cap y' = z \cap z'$ pentru un $z \in B$, atunci $x \cap y = x$.

4° Pentru toți $x, y \in B$, dacă $x \cap y = z$, atunci $x \cap y' = z \cap z'$ pentru toți $z \in B$.

5° B are cel puțin două elemente.

Se demonstrează că cele două definiții sînt echivalente.

Algebra B. Algebra booleană cu două elemente 0 și 1 va fi notată cu **B**. Ea este cea mai simplă algebră booleană nedegenerată, termen care desemnează proprietatea ca algebra să aibă cel puțin două elemente. (Algebra booleană formată dintr-un singur element a cu proprietatea de ordine $a \leqslant a$ este o algebră degenerată.)

Relațiile de ordine în **B** sînt $0 \leqslant 0, 0 \leqslant 1, 1 \leqslant 1$.

Operațiile se conformează următorului tablou :

x	y	$x \cup y$	$x \cap y$	$x \to y$
1	1	1	1	1
1	0	1	0	0
0	1	1	0	1
0	0	0	0	1

x	x'
0	1
1	0

Se observă că $x \leqslant y$ dacă și numai dacă $x \to y = 1$.

Toate proprietățile algebrelor booleene sînt verificate în **B**.

Algebre booleene atomice. Se numește *atom* al unei algebre booleene un element a astfel că dacă $a \neq 0$ și $b \leqslant a$, atunci $b = 0$ sau $b = a$.

Din definiție rezultă următoarele :

a) Dacă $x \neq 0$, există un atom a, astfel ca $a \leqslant x$.

b) Dacă a este un atom și x un element al algebrei, avem una din două situații : $a \leqslant x$ sau $a \cap x = 0$.

Întrucît $a \cap x \leqslant a$, rezultă $a \cap x = a$ sau $a \cap x = 0$, Dar ambele nu pot fi adevărate pentru că $a \neq 0$.

c) Dacă a este un atom și x un element al algebrei, avem una din două situații : $a \leqslant x$ sau $a \leqslant x'$ (enunț echivalent cu b).

Scanned with OKEN Scanner

Fie $A(x)$ mulțimea atomilor a astfel ca $a \leqslant x$. Presupunem că algebra booleană este finită. Proprietățile mulțimii $A(x)$ sînt :

1. $A(x \cap y) = A(x) \cap A(y)$.

Operația $\cap$ aplicată la mulțimi este reuniunea. Dacă $a \in A(x \cap y)$, $a \leqslant x \cap y$ și $a \leqslant x$, $a \leqslant y$, deci $a \in A(x) \cap A(y)$. Invers, dacă $a \in A(x)$ și $a \in A(y)$, $a \leqslant x$ și $a \leqslant y$ și $a \leqslant x \cap y$, $a \in A(x \cap y)$.

2. $A(x') = A(1) - A(x)$. (Notăm cu $-$ diferența de mulțimi).

Notăm cu 1 mulțimea tuturor atomilor lui B. Dacă $a \in A(x')$, atunci $a \leqslant x'$, iar $a \leqslant x$ nu este adevărată. Prin urmare, $a \in A(x)$ și $a \in A(1) - A(x)$. Invers, $a \in A(1) - A(x)$, $a \notin A(x)$, deci $a \in A(x')$.

3. $A(x) = A(y)$ dacă și numai dacă $x = y$.

Presupunem că $x \neq y$ și că $x \leqslant y$ nu este adevărată. Atunci $x \cap y' \neq 0$ și deci există $a \leqslant x \cap y'$. Atunci $a \in A(x)$ și $a \in A(y')$. $a \notin A(y)$, deci $A(x) \neq A(y)$. La fel cînd presupunem că $y \leqslant x$ nu este adevărată.

4. Dacă $a_1, \ldots, a_n$ sînt atomi distincți, $A(a_1 \cup \ldots \cup a_n) = \{a_1, \ldots, a_n\}$. Orice element din mulțimea din membrul al doilea al egalității se află și în $A(a_1 \cup \ldots \cup a_n)$. Invers, presupunem că x este în $A(a_1 \cup \ldots \cup a_n)$, dar nu și în $\{a_1, \ldots, a_n\}$, deci $x \neq a_i$. Atunci $x \cap a_i = 0$, $i = 1, \ldots, n$ și $x = x \cap (a_1 \cup \ldots \cup a_n) = 0$, contradicție.

Vom folosi aceste proprietăți în stabilirea teoremelor de reprezentare (v. 3.7).

O algebră booleană este *atomică* dacă pentru orice element x diferit de zero există un atom a al lui B, $a \leqslant x$.

Definiție. Un inel $R = (R, +, \cdot)$ cu unitate se numește *inel boolean* dacă pentru toți $x \in R$ există proprietatea $x^2 = x$.

Un inel este grup comutativ (abelian) față de operația $+$:

1) $x + y \in R$.

2) $(x + y) + z = x + (y + z)$.

3) $x + y = y + x$.

4) Ecuațiile $x + a = b$ și $a + x = b$, $a, b \in R$, au întotdeauna soluție, ceea ce asigură existența elementului neutru 0 și al inversului oricărui element, notat cu $-x$.

Operația $\cdot$ este definită de :

5) $xy \in R$,

6) $x(yz) = x(yz)$,

iar legătura între operații este dată de regulile de distributivitate :

7) $x(y + z) = xy + xz$,

8) $(x + y)z = xz + yz$.

Caracteristica unui inel R este cel mai mic număr natural k, pentru care $kx = \underbrace{x + x + \ldots + x}_{k \text{ ori}} = 0$, pentru toți $x \in R$.

Scanned with OKEN Scanner

Proprietatea booleană de idempotență $x^2 = x$ atrage următoarea proprietate:

Un inel boolean este comutativ față de $\cdot$ *și are caracteristica* 2.

Trebuie să demonstrăm deci, că într-un inel boolean $x + x = 0$.

Fie $x, y \in R$. Putem scrie

$$x + y = (x + y)(x + y) = x^2 + xy + yx + y^2.$$

Deci $xy + yx = 0$. Făcînd $x = y$, $x^2 + x^2 = 0$, avem

$$2x = x + x = x^2 + x^2 = 0.$$

Comutativitatea rezultă din $xy = xy + xy + yx = yx$, pentru că $xy + yx = 0$.

Proprietatea idempotenței există în latice și algebre booleene: $x \cap x = x$, dar proprietatea de inel boolean nu este valabilă: $x \cup x = x \neq 0$. Următoarea teoremă ne permite să dăm o structură de inel boolean unei algebre booleene și invers.

Teorema 1. *Dacă într-o algebră booleană* $B = (B, \cap, \cup, ')$ *definim*

$$x + y = (x \cap y') \cup (x' \cap y) \text{ și } xy = x \cap y,$$

obținem un inel boolean cu uniiate $R(B) = (B, +, \cdot)$.

Demonstrație. Vom arăta că proprietățile operațiilor de inel sînt satisfăcute. Comutativitatea lui $+$ rezultă din

$$x + y = (x \cap y') \cup (x' \cap y) = (y' \cap x) \cup (y \cap x') =$$
$$= (y \cap x') \cup (y' \cap x) = y + x.$$

Asociativitatea lui $+$:

$$(x + y) + z = ((x \cap y') \cup (x' \cap y)) + z =$$
$$= (((x \cap y') \cup (x'' \cap y)) \cap z') \cup (((x \cap y') \cup (x' \cap y))' \cap z) =$$
$$= (x \cap y' \cap z') \cup (x' \cap y \cap z') \cup (x \cap y \cap z) \cup (x' \cap y' \cap z) =$$
$$= (x' \cap ((y \cap z') \cup (y' \cap z))) \cup (x \cap ((y \cap z') \cup (y' \cap z))') = x + (y + z).$$

Pentru a încheia demonstrația că B formează grup față de $+$ rămîne să arătăm că

$$x + 0 = (x \cap 0') \cup (x' \cap 0) = x.$$

Proprietatea de inel boolean rezultă din

$$x + x = (x \cap x') \cup (x' \cap x) = 0.$$

Din definiția $xy = x \cap y$ rezultă imediat comutativitatea și asociativitatea lui $\cdot$ existente pentru $\cap$.

Distributivitatea lui $\cdot$:

$$xy + xz = (x \cap y) + (x \cap z) =$$
$$= ((x \cap y) \cap (x \cap z)') \cup ((x \cap y)' \cap (x \cap z)) =$$
$$= (x \cap y \cap (x' \cup z')) \cup ((x' \cup y') \cap x \cap z) =$$
$$= (x \cap y \cap z') \cup (x \cap z \cap y') = x \cap ((y \cap z') \cup (y' \cap z)) = x(y+z).$$

Scanned with OKEN Scanner

Rămîne să arătăm că $x1 = x \cap 1 = x$ și $x^2 = x \cap x = x$.

Concluzia este că algebra booleană $(B, \cap, \cup, ')$ a primit structura de inel cu unitate: $(B, +, \cdot, 0, 1)$.

Teorema 2. *Dacă într-un inel boolean $(R, +, \cdot)$ cu unitate definim $x \cup y = x + y + xy$ și $x \cap y = xy$, iar complementul lui x este $x' = x + 1$, obținem o algebră booleană $B(R) = (R, \cap, \cup, ')$.*

Comutivitatea celor două operații $\cap$, $\cup$ rezultă din definiții:

$$x \cup y = x + y + xy = y + x + yx = y \cup x,$$

$$x \cap y = xy = yx = y \cap x,$$

întrucît inelul boolean este comutativ.

Asociativitatea:

$$x \cup (y \cup z) = x \cup (y + z + yz) =$$

$$= x + y + z + yz + xy + xz + xyz = (x + y + xy) \cup z =$$

$$= (x \cup y) \cup z,$$

$$x \cap (y \cap z) = x(yz) = (xy)z = (x \cap y) \cap z.$$

Absorbția:

$$x \cap (x \cup y) = x(x + y + xy) = x + xy + xy = x + xy + xy + x,$$

$$x \cup (x \cap y) = x \cup (xy) = x + xy + xyx = x.$$

S-a demonstrat că $(B, \cap, \cup)$ este o latice.

Distributivitatea laticei $(B, \cap, \cup)$ se demonstrează imediat:

$$x \cap (y \cup z) = x(y + z + yz) = xy + xz + xyz = xy + xz + xyxz =$$

$$= xy \cup xz = (x \cap y) \cup (x \cap z).$$

Laticea $(B, \cap, \cup)$ este complementată, marginile inferioară și superioară fiind 0 și 1. Utilizăm definiția lui x' și anume $x' = 1 + x$. Avem

$$x \cap x' = x(1 + x) = x + x = 0,$$

$$x \cup x' = x + x' + xx' = x + 1 + x + x(1 + x) = 1 + x + x = 1.$$

Orice element x al laticei are complement x'.

Laticea $(B, \cap, \cup)$ fiind distributivă și complementată, este o algebră booleană.

Teorema 3. *O algebră booleană definită pe inelul boolean care corespunde acestei algebre este ea însăși o algebră booleană $B(R(B)) = B$. La fel $R(B(R)) = R$.*

Fie o algebră booleană B. Definim operațiile $\cap'$ și $\cup'$ pe $R(B)$:

$$x \cap' y = xy, \qquad x \cup' y = x + y + xy.$$

Atunci:

$$x \cap' y = xy = x \cap y,$$

$$x \cup' y = x + y + xy = 1 + (1 + x)(1 + y) =$$

$$= 1 + x'y' = (x' + y')' = x \cup y.$$

Scanned with OKEN Scanner

În ceea ce privește complementul :

$$1 + x = (1 \cap x') \cup (1' \cap x) = x' \cup 0 = x'.$$

Deci operațiile introduse pentru a trece de la inelul corespunzător unei algebre boolene la o algebră booleană sînt operațiile acestei algebre și $B(R(B)) = B$.

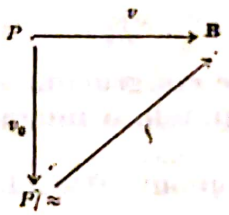
Se demonstrează la fel că $R(B(R)) = R$.

3.5. RELAȚIA DE CONGRUENȚĂ ȘI ALGEBRE BOOLEENE CÎT

În capitolul 2, teorema recurenței ne asigură de posibilitatea extinderii unei aplicații $v : P \to B$ la un omomorfism unic $\bar{v} : F \to B$, unde F este mulțimea liber generată de o clasă de funcții $\mathscr{F}$.

Teorema cu caracter mai general a fost ilustrată pe cazul care interesează logica și anume în care P este mulțimea variabilelor propoziționale, B algebra booleană **B**, F multimea formulelor, iar v funcția de evaluare a propozițiilor.

Se știe că funcția v determină pe mulțimea P o relație de echivalență, impărțind propozițiile între cele adevărate și false, adică între cele care au $v(p) = 1$ și cele care au $v(p) = 0$.



Toate propozițiile care au aceeași imagine în **B** sînt echivalente. Între mulțimea $P/\approx$ cu două clase de echivalență și algebra booleană cu două elemente $\{0, 1\}$ există o corespondență bijectivă. Ca funcții, v este surjecție, f bijecție. Funcția v_0 este aplicația naturală a mulțimii P pe $P/\approx$, care se numește și *factorizarea mulțimii P prin relația de echivalență* $\approx$. Funcția v face ca fiecărei propoziții p în P să-i corespundă în $P/\approx$ clasa $|p|$ a propozițiilor echivalente cu aceasta.

O relație de echivalență este o congruență față de o operație n-ară o dacă pentru elemente echivalente $(x_1, x_2, \ldots, x_n)$ și $(y_1, \ldots, y_n)$, unde $x_1 \approx y_1, \ldots, x_n \approx y_n$, rezultatul operației conduce la echivalență

$$o(x_1, \ldots, x_n) \approx o(y_1, \ldots, y_n).$$

În capitolul 2 am considerat două operații c și n corespunzînd implicației și negației ($\to$ și $\sim$) pentru că acestea au fost conectori adecvați ai sistemului deductiv considerat. În prezent vom ilustra considerațiile asupra congruenței pe operatorul unar $'$ și pe produs $\cap$.

Luăm ca punct de plecare algebra booleană $\langle B, \cap, ' \rangle$. Fie o congruență ω o relație de echivalență (reflexivă, simetrică, tranzitivă), care are proprietățile :

c_1) dacă $x \,\omega\, y$, atunci $x \cap z \,\omega\, y \cap z$ pentru orice z;

c_2) dacă $x \,\omega\, y$, atunci $x' \omega y'$.

Din acestea se deduce

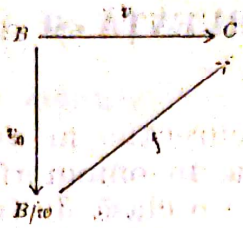
c_3) Dacă $x \omega y$ și $u \omega v$, atunci $x \cap u \omega y \cap v$.

Fie mulțimea de clase de echivalențe B/ω, ale cărei elemente sînt clasele $|x|$, $|y|$, ... Două clase sînt egale $|x| = |y|$ cînd $x \omega y$. Definim operații cu clasele menținînd pentru ușurință notațiile operațiilor din B:

$$|x| \cap |y| = |x \cap y|, \qquad |x|' = |x'|.$$

Se verifică că $\langle B/\omega, \cap, ' \rangle$ este o algebră booleană.

Fie C o altă algebră booleană $\langle C, \cap, ' \rangle$



Examinăm relațiile algebrei B/ω cu algebra din care a derivat prin utilizarea congruenței ω. Atunci v_0 este aplicația naturală, numită și *evaluare canonică*,

$$v_0 : B \to B/\omega,$$

unde $v_0(p) = |p|$. Ea este o surjecție care păstrează operațiile, deci un omomorfism:

$$v_0(x \cap y) = v_0(x) \cap v_0(y), \quad v_0(x') = (v_0(x))'.$$

Am stabilit că o relație de congruență ω determină un omomorfism și că unei congruențe îi corespunde o imagine omomorfică a lui B care este B/ω.

Reciproc, fiecărei imagini omomorfice a lui B îi corespunde o congruență în B.

Fie o altă algebră booleană C și o aplicație, care păstrează operațiile

$$v(x \cap y) = v(x) \cap v(y), \qquad v(x') = (v(x))';$$

deci C este imaginea omomorfică a lui B.

Relația ω definită astfel:

$$x \omega y \text{ dacă } v(x) = v(y),$$

este o relație de congruență, ale cărei proprietăți se verifică ușor. Se vede de asemenea că $f : B/\omega \to C$ definită cu

$$\langle |x|, v(x) \rangle, \quad |x| \in B/\omega$$

este un izomorfism și că

$$f(|x| \cap |y|) = f(|x \cap y|) = v(x \cap y) = v(x) \cap v(y) =$$
$$= f(|x|) \cap f(|y|),$$
$$f(|x'|) = v(x') = (v(x))' = (f(|x|))'.$$

Recapitulînd:

$$v_0 : B \to B/\omega, \; f_0 : B/\omega \to C, \; v : B \to C,$$

putem scrie descompunerea $v = f \cdot v_0$. Rezultatul poate fi enunțat astfel.

Scanned with OKEN Scanner

Fie B o algebră booleană și ω o relație de congruență pe B. Algebra B/ω a claselor de echivalență este imaginea omomorfică a lui B sub aplicația normală a lui B în B/ω. Invers, dacă algebra C este o imagine omomorfică a lui B, atunci C este izomorfă cu o algebră B/ω. Dacă $v : B \to C$ este omomorfismul respectiv, atunci $v = f \circ v_0$, unde $v_0 : B \to B/\omega$ și izomorfismul $f : B/\omega \to C$.

Menționăm că este vorba de *congruențe proprii*, adică diferite de relația universală $\langle x, x \rangle$.

Corespondența biunivocă între clasa tuturor imaginilor omomorfe ale lui B și ale tuturor congruențelor ce pot fi definite pe B se va întregi cu o altă corespondență biunivocă atît a congruențelor cît și a omomorfismelor cu mulțimea idealelor principale în B (v. 3.6).

Proprietățile stabilite pentru algebrele booleene au un caracter mai general. De aceea este util să parcurgem cîteva din noțiunile fundamentale ale algebrei universale.

O *algebră* este o structură $\langle A, o_1, \ldots, o_s \rangle$, unde A este o mulțime nespecificată, iar $o_1, \ldots, o_s$ operații unare, binare, ternare, ..., n-are. De multe ori, litera care desemnează mulțimea este luată drept denumire a algebrei. O *subalgebră* a lui A este o mulțime $A' \subseteq A$ închisă față de aceleași operații restrînse la A'.

Fiind dată o submulțime nevidă $A_0 \subseteq A$, intersecția tuturor subalgebrelor care o conțin este tot o subalgebră. Pentru orice A_0 există cea mică subalgebră A' care o conține și care se numește *algebra generată de* A_0, iar A_0 *mulțimea generatorilor lui* A'. Trecerea de la P (variabile propoziționale sau atomi) la F formule este legată de această noțiune.

Algebrele cu mulțimi de elemente diferite, dar avînd același număr de operații de aceeași aritate se numesc *similare*. Algebrele $\langle A, o_1, \ldots, o_n \rangle$ și $\langle B, o'_1, \ldots, o'_m \rangle$ sînt similare cînd $n = m$ și cînd pentru orice $i = 1, \ldots, n$ operațiile o_i și o'_i au același număr de argumente. Am considerat pînă acum algebre booleene similare, deși pentru demonstrații ne-am mărginit la doi operatori, ceilalți putînd fi definiți prin ei.

Enumerăm cîteva definiții și enunțuri din domeniul algebrei universale, pentru a situa mai bine rezultatele asupra algebrelor booleene.

1. Fiind date algebrele similare A, B, C și omomorfismele $f : A \to B$, $g : B \to C$, compunerea $h = g \circ f$, $h : A \to C$ este un omomorfism.

2. Fie A_0 mulțimeă de generatori a unei algebre A și $h_1 : A \to B$ și $h_2 : A \to B$ două omomorfisme într-o altă algebră similară B. Dacă $h_1(x) = h_2(x)$ pentru orice $x \in A_0$, atunci $h_1(x) = h(x_2)$ pentru orice $x \in A$.

Acest rezultat a fost folosit în 2.8 sub forma : cînd o aplicație $A_0 \to B$ poate fi extinsă la un omomorfism $A \to B$, atunci extensia este unică.

3. Fie aplicația $f : A_0 \to B_0$ a unei mulțimi de generatori a unei algebre A în altă mulțime de generatori B_0 a unei algebre similare B. Dacă f poate fi extins la un omomorfism $h : A \to B$ și f^{-1} poate fi extins la un omomorfism $g : B \to A$, atunci h este un izomorfism de la A la B și $h^{-1} = g$.

4. Se numește *algebră liberă* în clasa algebrelor similare o algebră A care îndeplinește condițiile :

a) conține o mulțime A_0 care generează pe A,

b) fiecare aplicație $A_0 \to B$, unde B este altă algebră în clasa algebrelor similare poate fi extinsă la un omomorfism $h : A \to B$.

Scanned with OKEN Scanner

Mulțimea A_0 se numește *mulțime a generatorilor liberi*.

Enunțul generalizează rezultatul tratat în 2.8.

Noțiunea de algebră liberă este esențială pentru logica matematică.

5. Fie $h : A \to B$ un omomorfism între algebrele A și B. Introducînd relația $x\omega y$ dacă și numai dacă $h(x) = h(y)$, definim o congruență ω în A. Algebra B este izomorfă cu A/ω și aplicația

$$h_0(|x|) = h(x) \text{ pentru } x \in A$$

este un izomorfism de la A/ω la B.

Este forma generală a teoremei demonstrate pentru algebrele booleene.

6. Fie ω o congruență în A și h un omomorfism de la A la o algebră similară B astfel că, dacă $x\omega y$, atunci $h(x) = h(y)$. Formula $h_0(|x|) = h(x)$ pentru $x \in A$ definește un omomorfism $h_0 : A/\omega \to B$.

Slăbind condițiile de definire a congruenței (lipsește dacă și numai dacă), h_0 nu mai este izomorfism.

Algebra Lindenbaum. Trecerea de la calculul propozițiilor pe limbajul $\mathscr{L}$ la o algebră $\mathscr{A}$ se face simplu : toate formulele limbajului devin mulțimea de elemente ale algebrei, iar conectorii limbajului de ordinul 0 : $\wedge$, $\vee$, $\to$, $\sim$ devin operatorii algebrei. Algebra asociată se numește *algebra formulelor* și are forma $\langle F, \wedge, \vee, \to, \sim\rangle$ sau $\langle F, \wedge, \sim\rangle$ sau $\langle F, \vee, \sim\rangle$ dacă sistemul logic se bazează pe o pereche de conectori adecvați.

Această algebră nu este booleană. Egalitatea pe care este construită mulțimea formulelor pretinde ca formulele să fie egale simbol cu simbol la examinarea parantezelor și $(A \wedge B)$ nu este egală cu $(B \wedge A)$. Aceste formule devin echivalente numai după ce se introduce o relație de echivalență (congruență) $\Leftrightarrow$ bazată pe funcția de evaluare și tablele de adevăr. Ne așteptăm că algebra cît $\langle F/\Leftrightarrow, \wedge, \vee, \to, \sim\rangle$ să fie algebră booleană. Se verifică ușor că această algebră este o algebră booleană avînd ca element 0 clasa $|A \wedge A'|$ și element 1 clasa $|A \vee A'|$. Elementul 1 este clasa tuturor tautologiilor, iar elementul 0 este clasa tuturor contradicțiilor. Două formule se găsesc în aceeași clasă dacă sînt echivalente, adică dacă au aceeași tablă de adevăr. O algebră de forma $\langle F/\Leftrightarrow, \wedge, \vee, \to, \sim\rangle$ se numește *algebra Lindenbaum* a calculului propozițiilor.

Condițiile teoremei privind obținerea unei algebre booleene cît B/ω dintr-o altă algebră booleană B nu corespund în întregime situației de față. Dar asigurîndu-ne că F este liber generată de o mulțime de generatori F_0 sau P și congruența introdusă satisface anumite condiții suplimentare, F/ω rezultă drept o algebră booleană.

Fie mulțimea de congruențe care satisfac următoarele condiții :

a) Dacă $x\omega y$, atunci $x \wedge z\omega y \wedge z$ pentru toți z.

b) Dacă $x\omega y$, atunci $x'\omega y'$.

c) $x \wedge y\omega y \wedge x$.

d) $x \wedge (y \wedge z)\omega(x \wedge y) \wedge z$.

e) Dacă $x \wedge y'\omega\ z \wedge z'$ pentru un z, atunci $x \wedge y\ \omega\ x$.

f) Dacă $x \wedge y\ \omega\ x$, atunci $x \wedge y'\omega\ z \wedge z'$ pentru toți z.

După cum se vede, s-au adăugat la cerințele a) și b) normale pentru o congruență condiții de comutativitate și asociativitate (c și d), precum și două condiții luate din axiomele 3 și 4 ale algebrei booleene.

Nu este corect să spunem : „calculul propozițiilor cu conectorii $\wedge$, $\vee$ și $\sim$ este o algebră booleană”, întrucît două mulțimi cu număr diferit

Scanned with OKEN Scanner

de variabile propoziționale generează două mulțimi diferite de formule cărora li se asociază două algebre booleene distincte. Elementele acestora nu mai sînt formule, ci clase.

3.6. IDEALE ȘI FILTRE

Definiții și proprietăți. O submulțime nevidă a unei algebre booleene $\Delta \subseteq B$ se numește *ideal* în B, dacă, oricare ar fi elementele ei $i_1, i_2 \in \Delta$ și un element oarecare b, $b \in B$:

$$i_1 \cap b \in \Delta, \qquad i_1 \cup i_2 \in \Delta.$$

0 și B sînt ideale ale lui B. Luînd $b = i_1'$, primim $0 \in I$; cazul $\Delta = B$ este inclus în definiție.

Idealele care diferă de 0 și B se numesc *ideale proprii*. Un ideal propriu nu poate conține pe 1, deoarece $b = 1 \cap b \in \Delta = B$.

Idealul unei algebre Boole $B = (B, \cap, \cup, 0, 1, ')$ este ideal al inelului boolean $(B, +, .)$.

Folosim definiția operațiilor $+$, $.$ în inel:

$$i_1 + i_2 = (i_1 \cap i_2') \cup (i_1' \cap i_2) \in \Delta \quad \text{și} \quad i_1 \cdot b = i_1 \cap b \in \Delta.$$

Definiția idealului Δ este echivalentă cu cerința ca pentru toți $i_1, i_2 \in \Delta$, $b \in B$ să avem

$$b \leqslant i_1 \Rightarrow b \in \Delta, \qquad i_1 \cup i_2 \in \Delta.$$

Folosim definiția $b \leqslant i_1 \Leftrightarrow b \cap i_1 = b \in \Delta$.

Mulțimea $\Delta(a) = \{x \cap a \mid x \in B\}$ este un ideal care se numește *ideal principal.*

Într-adevăr, fie $i_1 = x_1 \cap a$, $i_2 = x_2 \cap a$. Atunci

$$i_1 \cup i_2 = (x_1 \cup x_2) \cap a \text{ și } i_1 \cap b = (x_1 \cap a) \cap b = x_1 \cap (a \cap b).$$

Se întrebuințează expresia *idealului principal generat de elementul a.*

Un ideal Δ_m în B este *maximal* cînd singurele ideale în B care conțin pe Δ_m sînt Δ_m și B.

Cînd Δ este ideal al algebrei booleene B și deci al inelului boolean $(B, +, .)$, putem forma *inelul factorial* B/Δ. Acesta ne conduce din nou la o algebră booleană, pe care o notăm B/Δ.

Putem demonstra cu ajutorul propozițiilor date următoarea proprietate a unui ideal principal.

Idealul principal generat de un element $b \in B$ este

$$\Delta(a) = \{b \in B \mid b \leqslant a\}.$$

Pentru demonstrație ne folosim de definiția unui ideal principal și de proprietatea

$$b = x \cap a,\ b = b \cap a \text{ și } b \leqslant a \Leftrightarrow b \cap a = b.$$

Exemple. a) Fie $B = \mathcal{P}(M)$. Mulțimea $\{A \subseteq M \mid A \text{ finită}\}$ formează un ideal al lui B, care nu este principal.

b) Fie $B = \mathcal{P}(M)$. Mulțimea $\{N \mid N \subseteq A\}$ pentru oricare $A \in \mathcal{P}(M)$ formează un ideal principial în B.

Filtru sau *ideal dual* se numește submulțimea $\nabla \subseteq B$ pentru care, oricare ar fi $f_1, f_2 \in F$, iar b fiind un element oarecare al lui B, avem $f_1 \cup b \in \nabla$, $f_2 \cap f_1 \in \nabla$.

Scanned with OKEN Scanner

Definiția duală a filtrului ne permite să stabilim următoarea

Proprietate. *Fie* Δ, $\nabla \subseteq B$, *cînd* B *este o algebră booleană. Dacă* Δ *este ideal, mulțimea* $\Delta' = \{i' \mid i \in \Delta\}$ *este un filtru; dacă* ∇ *este filtru, mulțimea* $\nabla' = \{f' \mid f \in \nabla\}$ *este ideal.*

Este ușor de văzut că complementele elementelor unui ideal satisfac definiția filtrului:

$$i_1' \cap i_2' = (i_1 \cup i_2)' \in \Delta', \quad (i_1' \cup b) = (i_2 \cap b')' \in \Delta'$$

și invers.

Dualitatea ne permite să enunțăm o definiție alternativă a filtrului: $\Delta \subseteq B$ este filtru, dacă pentru toți $f_1, f_2 \in \nabla$, $b \in B$ avem

$$b \geqslant f_1 \Rightarrow b \in \nabla \quad \text{și} \quad f_1 \cap f_2 \in \nabla.$$

Ideale și congruențe. Fie ω o relație de congruență pe algebra booleană B.

Teoremă. *Mulțimea elementelor din* B *care sînt congruente cu* 0 *formează un ideal propriu în* B, $\Delta = \{x \in B \mid x\omega 0\}$ *și* $x\omega y$ *dacă și numai dacă* $x + y \in \Delta$. Reciproc, *dacă* Δ *este un ideal propriu al lui* B, *atunci relația* ω *introdusă astfel încît* $x\omega y$ *dacă și numai dacă* $x + y \in \Delta$ *este o relație de congruență și* $\Delta = \{x \in B \mid x\omega 0\}$.

(În enunț se utilizează notația $x + y = (x \cap y') \cup (x' \cap y)$ din teorema privind inelele booleene (v. 3.4), de fapt suma disjunctivă $\oplus$.)

Demonstrație. Fie ω o relație (proprie) de congruență în B și $\Delta = \{x \in B \mid x\omega 0\}$. Rezultă că $\Delta \subset B$. Dacă $x, y \in \Delta$, atunci $x\omega 0$, $x'\omega 1$, $x' \cap y'\omega 1 \cap y'$, $x' \cap y'\omega y'$ și $x \cup y\, \omega y$. Din $x \cup y\omega y$ și $y\omega 0$ rezultă $x \cup y\omega 0$. Fie $x \in \Delta$ și $z \in B$. Din $x\omega 0$ avem $x \cap z\omega 0$. Deci cele două condiții ale unui ideal sînt îndeplinite: $x \cup y \in \Delta$ și $x \cap z \in \Delta$ (unde $z \in B$).

Fie $x + y \in \Delta$, adică $x + y\omega 0$ și $(x + y) + y\omega 0 + y$. Dar $(x + y) + y = x$ și deci $x\omega y$. Dacă $x\omega y$, atunci $x + y\omega y + y$ și $x + y\omega \mathbf{0}$. Au fost folosite proprietățile sumei ce se obțin imediat prin substituția $x + y = (x \cap y') \cup (x' \cap y)$.

Reciproca teoremei se demonstrează arătînd că ω, definită prin $x + y \in \Delta$, este o relație reflexivă, simetrică și tranzitivă și că este o congruență față de operațiile $\cap$ și $'$, $x \cap z\omega y \cap z$ și $x'\omega y'$. Faptul că $x\omega 0$ dacă și numai dacă $x \in \Delta$ rezultă din $x + 0 = x$. Interesul acestei teoreme rezidă în faptul că relațiile de congruență (proprii) ale algebrei booleene B sînt puse în corespondență biunivocă cu idealele proprii ale lui B.

Făcînd legătura cu teorema care stabilește corespondența biunivocă a congruențelor lui B cu imaginile sale omomorfice printr-un izomorfism, putem asocia la fiecare astfel de imagine un ideal propriu Δ. Dacă $h : B \to C$ este un omomorfism, idealul Δ asociat se va numi *nucleul omomorfismului* h. El este mulțimea elementelor lui B pe care h le aplică în elementul zero al lui C. În loc să scriem B/ω pentru imaginea izomorfă a lui C, scriem de obicei B/Δ, unde Δ este idealul asociat congruenței.

Dacă h este un izomorfism, atunci ω este egalitatea în B și Δ este *idealul zero*. Dacă omomorfismul h este idealul zero, h este un izomorfism.

Scanned with OKEN Scanner

Putem enunța proprietatea : *un omomorfism este un izomorfism dacă și numai dacă nucleul său este idealul zero.*

Ideale maximale. Un *ideal maximal* este un ideal propriu Δ, $\Delta \subset B$ și $1 \notin \Delta$, care nu este submulțime proprie a unui ideal propriu, deci este maximal în mulțimea ordonată a idealelor proprii.

Lema lui Zorn (v. 3.1) aplicată la această mulțime ne asigură existența unui ideal maximal în B.

Orice ideal propriu este cuprins într-un ideal maximal.

Într-adevăr, considerăm mulțimea ordonată a idealelor proprii $(P, \subseteq)$. Fie o submulțime total ordonată L a lui P. Atunci mulțimea M a elementelor din B care se găsesc în unele Δ din L este o *margine superioară* pentru L:

$$M = \{x \in B \mid x \in \Delta \text{ pentru unele } \Delta \text{ din } L\}.$$

Verificăm că M este un ideal și pentru că $1 \notin M$ (nu aparține nici unui membru al lui L), $M \in P$. Sîntem în condițiile aplicării lemei lui Zorn : *orice submulțime total ordonată (lanț) în P are o margine superioară în P, deci P are element maximal.*

Fie acum $\{\Delta \in P \mid \Delta \supseteq \Delta_1\}$ mulțimea idealelor diferite de B care includ pe Δ_1. Același raționament aplicat acestei mulțimi ne asigură de existența unui ideal maximal, care include pe Δ_1.

Rezultă două proprietăți importante ale idealelor maximale, într-o algebră booleană :

a) Pentru orice $x \neq 1$, există un ideal maximal Δ_m astfel ca $x \in \Delta_m$.

Se demonstrează luînd x drept idealul (x) și afirmînd existența lui $\Delta_m \supseteq (x)$.

b) Pentru fiecare element $x \in B$ și fiecare ideal maximal Δ_m, numai una din cele două relații este valabilă : $x \in \Delta_m$ sau $x' \in \Delta_m$.

Într-adevăr, ambele nu pot fi simultan valabile pentru că ar însemna că $x \cup x' = 1 \in \Delta_m$, ceea ce nu este posibil. Să arătăm că dacă $x \notin \Delta_m$, rezultă $x' \in \Delta_m$.

Fie mulțimea C a elementelor lui B de forma $a \cup b$, unde $a \leqslant x$ și $b \in \Delta_m$. Mulțimea C este un ideal pentru că

$$(a_1 \cup b) \cup (a_2 \cup b) = (a_1 \cup a_2) \cup (b_1 \cup b_2) = a_3 \cup b_3$$

și dacă $y \in B$, $(a \cup b) \cap y = (a \cap y) \cup (b \cap y) = a_1 \cup b_1$. Pe de altă parte, $\Delta_m \subset C$ pentru că, conform construcției, $\Delta_m \subseteq C$ și $x \in C$, dar $x \notin \Delta_m$; idealul Δ_m este maximal și deci $C = B$. Pentru un anume a, $a \leqslant x$ și $b \in \Delta_m$, $a \cup b = 1$. Rezultă că $x \cup (a \cup b) = x \cup 1$ sau $x \cup b = 1$ pentru că $a \leqslant x$. Putem scrie $x' = x' \cap 1 = x' \cap (x \cup b) = (x' \cap x) \cup (x' \cap b) = x' \cap b$. Fiind de forma $x' \cap b$, unde $b \in \Delta_m$, $x' \in \Delta_m$, ceea ce era de demonstrat.

În analogie cu mulțimile $A(x)$ (mulțimile atomilor pentru care $a \leqslant x$) introducem mulțimile $M(x)$, mulțimea tuturor idealelor maximale Δ_m astfel ca $x \notin \Delta_m$ sau echivalent, conform cu b), $x' \in \Delta_m$.

Se demonstrează următoarele proprietăți :

1) $M(x \cap y) = M(x) \cap M(y)$;

2) $M(x') = M(1) - M(x)$, unde $M(1)$ este mulțimea tuturor idealelor maximale ale unei algebre;

3) $M(x) = M(y)$ dacă și numai dacă $x = y$.

Vom demonstra 1). Fie $\Delta_m \in M(x \cap y)$. Atunci $(x \cap y)' = x' \cup y' \in \in \Delta_m$. Dar $x' = x' \cap (x' \cup y')$ și $y' = y' \cap (x' \cup y')$ (absorbția); urmează că $x' \in \Delta_m$ și $y' \in \Delta_m$. Deci $\Delta_m \in M(x)$ și $\Delta_m \in M(y)$ sau $\Delta_m \in M(x) \cap \cap M(y)$. Reciproca se demonstrează la fel.

3.7. TEOREME DE REPREZENTARE

Mulțimea $\mathscr{P}(M)$ a părților unei mulțimi M, pe care sînt definite operațiile de reuniune $\cup$ și intersecție $\cap$, are structură laticeală. Latticea pe care o constituie $\mathscr{P}(M)$ este distributivă și complementată (dacă introducem operația de complementaritate).

Algebra $\langle \mathscr{P}(M), \cap, \cup, ' \rangle$ este o algebră booleană. Următoarele teoreme au adîncit legătura dintre algebrele de mulțimi și algebrele booleene:

Teorema 1. Fie B o algebră booleană cu n elemente. *Aceasta este izomorfă cu algebra submulțimilor mulțimii de atomi ai lui B. Dacă B are m atomi, $n = 2^m$.*

Fie B_0 mulțimea atomilor lui B. Aplicația $f: B \to \mathscr{P}(B_0)$ și $x \to A(x)$, unde $A(x)$ este mulțimea definită la 3.4, este o bijecție conform cu proprietatea 3 a mulțimilor $A(x)$. Proprietatea 1 ne asigură că $f(x \cap y) = f(x) \cap \cap f(y)$, iar proprietatea 2 că $f(x') = (f(x))'$, deci f este un izomorfism.

Teorema 2. *Fie B o algebră booleană. Ea este izomorfă cu algebra mulțimilor, bazată pe mulțimea tuturor idealelor maximale ale lui B.*

Formăm mulțimea M a tuturor idealelor de forma $M(x)$, unde $M(x)$ are definiția dată în (3.6). Aplicația $g: B \to M$ și $x \to M(x)$ este o bijecție prin proprietatea 3 și un izomorfism, pentru că $g(x \cap y) = g(x) \cap g(y)$ și $g(x') = (g(x))'$ conform proprietăților 1 și 2 ale mulțimilor $M(x)$.

Teorema 3. *Pentru ca o algebră booleană B să fie izomorfă cu algebra submulțimilor unei mulțimi este necesar și suficient ca B să fie completă și atomică. În acest caz, B este izomorfă cu algebra submulțimilor mulțimii atomilor săi.*

Condiția ca B să fie atomică este cunoscută din 3.4: B este atomică dacă și numai dacă pentru orice element $x \neq 0$ există un atom a din B cu $a \leqslant x$.

Algebra booleană se numește *completă* dacă și numai dacă pentru fiecare submulțime A a lui B există $\sup A$ relativ la relația de ordine parțială. Este evident că orice *algebră booleană finită este completă*. Orice pereche de elemente are cea mai mică margine superioară, notată $\sup(a, b)$. Deci pentru orice submulțime de elemente, fiind finită, există $\sup A$. Cînd B are o infinitate de elemente, nu este sigur că $\sup A$ (unde $A \subset B$) există. Completitudinea asigură acest lucru.

Proprietatea completitudinii se definește și pentru latice sub forma: dacă o latice are elementul 0 și $\bigcup L'$, suma elementelor lui L' există

Scanned with OKEN Scanner

pentru fiecare submulțime $L' \subset L$, atunci L este completă. Laticea numerelor reale este completă și laticea numerelor raționale nu este completă.

Algebra submulțimilor unei mulțimi este atomică și completă și aceste proprietăți se păstrează într-un izomorfism.

Condițiile enunțate de teoremă sînt deci necesare. Să arătăm că sînt și suficiente. Fie B o algebră atomică și completă și B_0 mulțimea atomilor săi. Ca și în cazul finit, numim $A(x)$ mulțimea atomilor a pentru care $a \leqslant x$. Aplicația $f : B \to \mathscr{P}(B_0)$ are proprietățile 1 și 2, deci este un omomorfism de la B la algebra submulțimilor lui B_0. Dacă C este o mulțime oarecare a lui B_0, atunci C are sup $C = c$ în B conform ipotezei completitudinii. Urmează că $f(a) = C$, conform generalizării lui A, pentru cazul infinit. Deci, f este o aplicație în $\mathscr{P}(B_0)$. Faptul că este bijectivă rezultă din faptul că pentru $x \neq 0$, $A(x) \neq 0$ și $A(x) = 0$ dacă și numai dacă $x = 0$. A fost folosită atomicitatea lui B.

Un corolar al teoremei 1 se enunță astfel:

Două algebre booleene cu același număr finit de elemente sînt izomorfe.

3.8. POLINOAME BOOLEENE

Fie $X = \{x_1, \ldots, x_n\}$ o mulțime de simboluri, numite *variabile*. Un polinom boolean este definit recursiv astfel:

1) $x_1, \ldots, x_n$, 0 și 1 sînt polinoame booleene,

2) dacă p și q sînt polinoame booleene, atunci $(p \cap q)$, $(p \cup q)$, p' sînt polinoame booleene.

Se poate observa analogia între introducerea recursivă a formulei logice și aceea a polinomului boolean. Paralelismul continuă între funcția de adevăr și funcția polinomului boolean, între formele normale ale formulelor logice și ale polinoamelor booleene.

Definiție. *Funcția polinomială booleană* este o aplicație a produsului direct a n elemente din B pe B:

$$B^n \to B, \ (a_1, a_2, \ldots, a_n) \to \bar{p}_B(a_1, \ldots, a_n).$$

În cele ce urmează vom considera $B = \mathbf{B} = \{0, 1\}$ și $B^n = \mathbf{B}^n$, deși definiția se aplică la o algebră booleană oarecare B.

Echivalența a două polinoame $p \approx q$ se definește prin egalitatea funcțiilor polinomiale booleene $\bar{p}_\mathbf{B} = \bar{q}_\mathbf{B}$.

Exemplu. $X = \{x_1, x_2\}$. Polinoame booleene pe X sînt: 0, 1, x_1, x_2, $x_1 \cap x_2$, $x_2 \cap x_1$, $x_1 \cup x_2$, $x_2 \cup x_1$, x_1', x_2', $x_1' \cap x_2$, $x_2 \cap x_1'$ etc. Două polinoame sînt egale cînd secvențele de simboluri sînt identice, $x_1 \cap x_2$ nu este egal cu $x_2 \cap x_1$. Mulțimea polinoamelor formate pe X o numim P_2, și P_2 nu este o algebră booleană.

Cînd mulțimea polinoamelor este formată pe X cu cel mult n elemente, o notăm P_n. Se observă că

$$P_1 \subset P_2 \subset P_3 \ldots$$

Fie polinomul $p_1 \in P_2 : p_1 = x_1 \cap x_2$ și $\bar{p}_{1\mathbf{B}} : \mathbf{B}^2 \to \mathbf{B}$ este funcția polinomială:

$$(1,1) \to 1, \ (1,0) \to 0, \ (0,1) \to 0, \ (0,0) \to 0.$$

Fie polinomul $p_2 = x_2 \cap x_1$. Funcția polinomială $\bar{p}_{2\mathbf{B}}$ este

$$(1,1) \to 1, \ (1,0) \to 0, \ (0,1) \to 0, \ (0,0) \to 0.$$

Scanned with OKEN Scanner

Deși nu sînt egale, polinoamele p_1 și p_2 au funcții polinomiale identice, deci sînt *echivalente*.

O formă normală are următoarele proprietăți :

a) orice polinom este echivalent cu un polinom scris în formă normală,

b) orice două polinoame în formă normală distinctă corespund la două polinoame neechivalente.

Forma normală *sumă de produse* FND are expresia

$$p_{\cup} = \bigcup d_i(x_1 \cap x_2 \cap \ldots \cap x_n),\ d_i \in \{0, 1\}.$$

Forma normală *produs de sume* FNC are expresia

$$p_{\cap} = \bigcap c_i(x_1 \cup x_2 \cup \ldots \cup x_n),\ c_i \in \{0, 1\}.$$

În ambele forme normale x_i pot fi complementate sau nu.

Se demonstrează că oricărui polinom oarecare îi corespunde cîte un polinom unic determinat de forma de mai sus.

Aducerea la forma normală sumă de produse. Se aplică următoarele reguli :

P a s u l 1. Se aplică legile De Morgan și se aduce semnul complementarității după variabile și constante.

P a s u l 2. Se aplică legile asociativității pentru a exprima polinomul sub forma de sumă de produse.

P a s u l 3. Dacă o variabilă x_i lipsește dintr-un produs, se introduce $x_i \cup x_i'$.

P a s u l 4. Se aplică din nou distributivitatea, legea idempotenței și adunarea variabilelor.

Întrucît funcția booleană se scrie deseori sub formă de tabel analog tablei de adevăr, aceasta oferă un procedeu facil pentru scrierea formelor normale ale unui polinom.

Exemple. a) Fie $p = (x_1 \cap (x_2 \cup x_3)')' \cup ((x_1 \cap x_2) \cup x_3') \cap x_1$.

Calculăm direct

$\bar{p}_B(1, 1, 1) = 1$, $\bar{p}_B(0, 1, 1) = 0$,

$\bar{p}_B(1, 1, 0) = 1$, $\bar{p}_B(0, 1, 0) = 0$,

$\bar{p}_B(1, 0, 1) = 0$, $\bar{p}_B(0, 0, 1) = 0$,

$\bar{p}_B(1, 0, 0) = 1$, $\bar{p}_B(0, 0, 0) = 0$,

$p_{\cup} = (x_1 \cap x_2 \cap x_3) \cup (x_1 \cap x_2 \cap x_3') \cup (x_1 \cap x_2' \cap x_3')$.

b) Fie $p = (x_1 \cup x_2 \cup x_3) \cap ((x_1 \cap x_3) \cup (x_1' \cap x_3))'$. Avem

$\bar{p}_B(1, 1, 1) = 0$, $\bar{p}_B(0, 1, 0) = 0$

$\bar{p}_B(1, 1, 0) = 0$, $\bar{p}_B(0, 1, 0) = 1$,

$\bar{p}_B(1, 0, 1) = 1$, $\bar{p}_B(0, 0, 1) = 0$,

$\bar{p}_B(1, 0, 0) = 1$, $\bar{p}_B(0, 0, 0) = 0$,

$p_{\cap} = (x_1' \cup x_2' \cup x_3') \cap (x_1' \cup x_2' \cup x_3) \cap (x_1 \cup x_2' \cup x_3') \cap$

$\cap (x_1 \cup x_2 \cup x_3') \cap (x_1 \cup x_2 \cup x_3)$.

Scanned with OKEN Scanner

3.9. OMOMORFISME ȘI IZOMORFISME

Definiții. O funcție (aplicație) $f: L_1 \to L_2$, unde L_1 și L_2 sînt latice, se numește *omomorfism de sumă*, dacă

$$x \cup y = z \Rightarrow f(x) \cup f(y) = f(z), \quad \forall x,y,z \in L_1;$$

omomorfism de produs, dacă

$$x \cap y = z \Rightarrow f(x) \cap f(y) = f(z), \quad \forall x,y,z \in L_1;$$

omomorfism de ordine, dacă

$$x \leqslant y \Rightarrow f(x) \leqslant f(y), \quad \forall x, y, z \in L_1,$$

omomorfism (de latice), dacă este omomorfism de sumă și omomorfism de produs (vezi definițiile omomorfismului de la 2.8).

Omomorfismele de latice primesc următoarele denumiri după cum sînt:

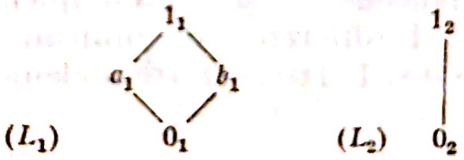
injective $\Leftrightarrow$ monomorfisme de latice,
surjective $\Leftrightarrow$ epimorfisme de latice,
bijective $\Leftrightarrow$ izomorfisme de latice.

Între funcțiile introduse pe latice se stabilesc următoarele relații:

Izomorfism $\Rightarrow$ Omomorfism $\Rightarrow$ (Omomorfism de sumă / Omomorfism de produs) $\Rightarrow$ Omomorfism de ordine

Un omomorfism (de sumă sau de produs) este un omomorfism de ordine, dar reciproca nu este adevărată.

Exemple. a) Fie L_1 și L_2 două latice reprezentate prin diagramele Hasse



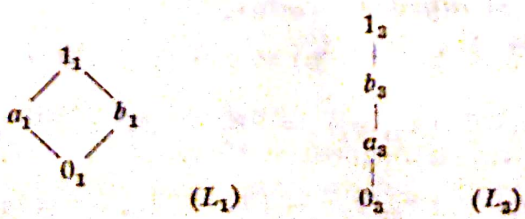
Fie funcția (surjectivă) definită astfel:

$f: L_1 \to L_2$, $f(0_1) = f(a_1) = f(b_1) = f(0_2) = 0_2$, $f(1_1) = 1_2$. Aceasta este un omomorfism de ordine: $a_1 \leqslant 1_1 \Rightarrow f(a_1) \leqslant f(a_2)$ pentru că $0_2 \leqslant 1_2$ etc.

Funcția f este un omomorfism de produs, deoarece $a_1 \cap b_1 = 0 \Rightarrow f(a_1) \cap f(b_1) = 0_2 = f(0_1) = f(a_1 \cap b_1)$; iar f nu este un omomorfism de sumă, deoarece $a_1 \cup b_1 = 1$, dar $f(a_1) \cup f(b_1) = 0_2 = f(a_1 \cup b_1) = f(1_1) = 1_2$.

Deci f nu este un omomorfism de latice.

b) Fie laticele L_1, L_3 definite de diagramele Hasse



și funcția $g(0_1) = 0_3$, $g(a_1) = a_3$, $g(b_1) = b_3$, $g(1_1) = 1_3$. Se observă că g este un omomorfism de ordine: $b_3 \leqslant 1_3 \Rightarrow g(b_1) \leqslant g(1_1)$ pentru că $b_3 \leqslant 1_3$ etc., iar g nu este un omomorfism

Scanned with OKEN Scanner

de produs :

$$a_1 \cap b_1 = 0, \quad g(a_1 \cap b_1) = 0, \quad g(a_1) \cap g(b_1) = a_3 \cap b_3 = a_3,$$

și g nu este un omomorfism de sumă :

$$a_1 \cup b_1 = 1_1, \quad g(a_1 \cup b_1) = 1_3, \quad g(a_1) \cup g(b_1) = a_3 \cup b_3 = b_3.$$

Omomorfism boolean. Un *omomorfism boolean* $f : B_1 \to B_2$, B_1 și B_2 fiind două algebre booleene, este definit ca un omomorfism de latice (în raport cu cele două operații $\cap$, $\cup$), la care adăugăm condiția $f(x') = (f(x))'$.

Notăm un omomorfism boolean $B_1 \cong B_2$.

Consecințe. Fie f un omomorfism boolean. Avem :

a) $f(0 \cup 1) = f(0) \cup f(1) = f(1)$ și deducem $f(0) = 0$, iar din $f(x') = (f(x))'$, unde $x = 0$, $f(1) = (f(0))' = 1$. Deci în orice omomorfism boolean $f(0) = 0$, $f(1) = 1$.

b) $f(B_1) = \operatorname{Im} f$ este o algebră booleană și o subalgebră a lui B_2. Arătăm că dacă $x, y \in B_1$, unde B_1 este o algebră booleană, atunci $f(x), f(y) \in f(B_1)$ formează o algebră booleană, iar $f(B_1)$ fiind o submulțime a lui B_2, închisă față de $\cap$, $\cup$,$'$ este o subalgebră a lui B_2.

Fie $x \in B_1$, $y \in B_1$, $x \cap y \in B_1 \Rightarrow f(x) \in f(B_1)$, $f(y) \in f(B_1) \Rightarrow f(x) \cap f(y) = f(x \cap y) \in f(B_1)$.

La fel $f(x) \cup f(y) = f(x \cup y) \in f(B_1)$, iar $(f(x))' = f(x') \in f(B_1)$.

Nucleul unui omomorfism boolean se notează și se definește astfel :

$$\operatorname{Ker} f = \{b \in B_1 \mid f(b) = 0\}.$$

Imaginea nucleului unei algebre booleene într-un omomorfism este 0.

Omomorfismele ce se stabilesc între algebre booleene de un interes particular și anume algebra mulțimii părților unei mulțimi M (studiate în 3.7) și algebra produs conduc la următoarele rezultate :

1. Fie M o mulțime și $\mathscr{P}(M)$ mulțimea părților sale (mulțimea submulțimilor sale). Este ușor de verificat că $B = (\mathscr{P}(M), \cap, \cup, \emptyset, ')$ este o algebră booleană. Considerăm operațiile pe mulțimi intersecția și reuniunea drept operațiile booleene $\cap$ și $\cup$. Complementul unei mulțimi $A \in \mathscr{P}(M)$ este $A' = M - A$ (diferență de mulțimi), iar $\emptyset$ și M joacă rolul marginilor universale 0 și 1. Dacă M are n elemente, cardinalul lui B este $|B| = 2^n$.

Observație. Sînt date două mulțimi M, N atfel ca $M \subset N$. Definim $f : \mathscr{P}(M) \to \mathscr{P}(N)$ și $f(A) = A$. Cele două mulțimi $\mathscr{P}(M)$ și $\mathscr{P}(N)$ cu definiția de mai sus formează algebre booleene. Funcția f este în mod evident un omomorfism de latice (f este injectivă, deci un monomorfism), dar f nu este omomorfism boolean, pentru că complementele oricărei mulțimi A, $A \in \mathscr{P}(M)$, sînt diferite în M și N :

$$f(A') = A' = M - A, \qquad (f(A))' = N - f(A) = N - A.$$

2. Fie B_1 și B_2 două algebre booleene. Mulțimea produselor $\{x, y) \mid x \in B_1, y \in B_2\}$ este produsul direct al lui B_1 și B_2 și se notează $B_1 \times B_2$, numindu-se și *algebra produs*.

Definim operațiile $\cap$, $\cup$, $'$:

$$(x_1, y_1) \cap (x_2, y_2) = (x_1 \cap x_2, y_1 \cap y_2),$$

$$(x_1, y_1) \cup (x_2, y_2) = (x_1 \cup x_2, y_1 \cup y_2),$$

$$(x, y)' = (x', y'), \; 0 = (0, 0), \; 1 = (1, 1).$$

Scanned with OKEN Scanner

Se verifică ușor că $B_1 \times B_2$ este o algebră Boole.

Pornim de la algebra Boole cu două elemente : $\mathbf{B} = \{0, 1\}$. Formăm produsul direct $\mathbf{B} \times \mathbf{B} \times \mathbf{B} \times \ldots \times \mathbf{B}$ (de n ori). Elementele acestei mulțimi vor fi secvențele de lungime n formate din 0 și 1 : $(e_1, e_2, \ldots, e_n)$, unde $e_i \in \{0, 1\}$.

Se demonstrează că $B = \langle \{0, 1\}^n, \cap, \cup, 0, 1, ' \rangle$ este o algebră booleană, unde operațiile au fost definite ca și în cazul produsului direct a două algebre booleene diferite :

$$(e_1, \ldots, e_n) \cap (i_1, \ldots, i_n) = (e_1 \cap i_1, \ldots, e_n \cap i_n),$$

$$(e_1, \ldots, e_n) \cup (i_1, \ldots, i_n) = (e_1 \cup i_1, \ldots, e_n \cup i_n).$$

$$(e_1, \ldots, e_n)' = (e_1', \ldots, e_n'), \; 0' = 1, \; 1' = 0,$$

$$0 = (0, \ldots, 0), \; 1 = (1, \ldots, 1).$$

Izomorfismul boolean între B și $\mathscr{P}(B_0)$ a fost demonstrat în 3.7.

Teoremă. *Pentru orice algebră booleană finită există* $n \in \mathbb{N}$ *astfel ca să existe izomorfismul*

$$B \cong \{0, 1\}^n.$$

Demonstrație. Procedăm prin inducție după cardinalul lui B. Dacă $|B| = 2$, teorema este evident satisfăcută. Presupunem $|B| > 2$ și fie $a \in B$, $a \neq 0$, $a \neq 1$. Mulțimea intervalelor $[0, a]$ este o algebră booleană B_1 : mulțimea intervalelor $[a, 1]$ este o algebră booleană B_2. Fie produsul direct $B_3 = B_1 \times B_2$. Aceasta este tot o algebră booleană. Folosim funcția $f : B \to B_3$, $f(b) = (a \cap b, \; a \cup b)$. Cînd $b = 0$, $f(b) = (0, a)$, iar cînd $b = 1$, $f(b) = (a, 1)$. În B_3, care va avea forma $[0, a] \times [a, 1] = (x, y)$, operația asupra elementelor se face ca cea definită anterior.

(1) $\quad (x_1, y_1) \cap (x_2, y_2) = (x_1 \cap x_2, y_1 \cap y_2),$

(2) $\quad (x_1, y_1) \cup (x_2, y_2) = (x_1 \cup x_2, y_1 \cup y_2).$

Conform definiției lui f :

$$f(b \cap c) = (a \cap (b \cap c), \; a \cup (b \cap c)).$$

Conform formulei (1) :

$$f(b) \cap f(c) = (a \cap b, \; a \cup b) \cap (a \cap c, \; a \cup c) =$$

$$= (a \cap b \cap c, \; (a \cup b) \cap (a \cup c)) = (a \cap b \cap c, \; a \cup (b \cap c)).$$

Deci $f(b \cap c) = f(b) \cap f(c)$. La fel

$$f(b \cup c) = (a \cap (b \cup c), \; a \cup (b \cup c)) = f(b) \cup f(c).$$

Pe de altă parte

$$f(b') = (a \cap b', \; a \cup b') = (a \cap b, \; a \cup b)' = (f(b))'.$$

Deci f este un omomorfism boolean.

Fie $(x, y) \in B_3$. Atunci (x, y) este imaginea lui b, adică $(x, y) = f(b)$, unde $b = y \cap (a' \cup x)$ este unic determinat. Deci f este bijecție și izomorfism. Am stabilit $B \cong B_3$.

Pe de altă parte, pe baza inducției, există n astfel ca $[0, a] \cong \{0, 1\}^n$ și există m astfel ca $[a, 1] \cong \{0, 1\}^m$. Deci $B \cong \{0, 1\}^{n+m}$.

Scanned with OKEN Scanner

Rezultă izomorfismul între $\{0, 1\}^n$ și $\mathscr{P}(M)$, unde $M = \{a_1, \ldots, a_n\}$.

Acest lucru poate fi utilizat pentru a stabili un izomorfism între B, o algebră booleană finită oarecare, $\{0, 1\}^n$ și $\mathscr{P}(A)$, unde A este mulțimea atomilor lui B, demonstrat în teorema de reprezentare 1 din 3.7.

Mulțimea funcțiilor $f: B^n \to B$ care aplică produsul cartezian al unei algebre booleene pe ea însăși formează o algebră booleană.

Definim operațiile pe funcții după cum urmează:

$$\begin{aligned} h &= f \cap g, & h(x) &= f(x) \cap g(x), \\ k &= f \cup g, & h(x) &= f(x) \cup g(x), \\ l &= f', & l(x) &= (f(x))', \\ f_0, & & f_0(x) &= 0, \\ f_1, & & f_1(x) &= 1, \end{aligned}$$

pentru toți $x \in B^n$. Mulțimea tuturor funcțiilor $f: B^n \to B$, pe care o notăm $\mathscr{F}_n(B)$, este o algebră booleană.

Putem acum să stabilim o legătură cu funcțiile polinomiale booleene. Într-adevăr, o funcție polinomială face să corespundă fiecărui polinom p o funcție polinomială (care corespunde funcției de adevăr a formulelor logice). Funcția $\bar{p}_B(a_1, \ldots, a_n)$ este o aplicație $B^n \to B$. Notăm $P_n(B) = \{\bar{p}_B \mid p \in P_n\}$.

Teoremă. *Fie B o algebră booleană. Mulțimea $P_n(B)$ este o algebră booleană și o subalgebră a algebrei booleene $\mathscr{F}_n(B)$ a tuturor funcțiilor $f: B^n \to B$.*

Demonstrație. Avem

$(\bar{p}_B \cap \bar{q}_B)(a_1, \ldots, a_n) = \bar{p}_B(a_1, \ldots, a_n) \cap \bar{q}_B(a_1, \ldots, a_n) = p_B \cap q_B(a_1, \ldots, a_n)$ pentru orice $a_i \in B$. Aceasta înseamnă că, oricare ar fi $\bar{p}_B$ și $\bar{q}_B \in P_n(B)$, $\bar{p}_B \cap \bar{q}_B = \overline{p_B \cap q_B} \in P_n(B)$. Deci $P_n(B)$ este închisă față de $\cap$. La fel se arată că este închisă față de $\cup$ și $'$. Definim $\bar{0} = f_0$ și $\bar{1} = f_1$.

Am definit două polinoame echivalente $p, q \in P_n$ prin

$$p \approx q \Leftrightarrow \bar{p}_B = \bar{q}_B,$$

deci prin egalitatea funcțiilor booleene pe B. Următoarele propoziții arată că:

a) relația $\approx$ care este o relație de echivalență împarte polinoamele în clase de polinoame, având aceeași funcție polinomială;

b) relația $\approx$ se menține pe o algebră booleană oarecare, nefiind nevoie de o nouă definiție de echivalență pe aceasta, pentru că ea atrage $\bar{p}_B = \bar{q}_B$.

Relația $\approx$ este o relație de echivalență.

Verificăm reflexivitatea: $p \approx p$. Evident: $\bar{p}_B = \bar{p}_B$.

Simetria: $p \approx q \Rightarrow q \approx p$, pentru că $\bar{p}_B = \bar{q}_B \Rightarrow \bar{q}_B = \bar{p}_B$, oricare ar fi $p_B, q_B \in P_n$.

Tranzitivitatea: $p \approx q$, $q \approx r \Rightarrow p \approx r$ pentru că $\bar{p}_B = \bar{q}_B$, $\bar{q}_B = \bar{r}_B \Rightarrow \bar{p}_B = \bar{r}_B$.

$P_n/\approx$ este o algebră booleană în raport cu operațiile pe clasele de echivalență și este izomorfă cu $P_n(B)$.

Formăm funcția $f: P_n(B) \to P_n/\approx$, care aplică $\bar{p}_B$ pe clasa de echivalență cu polinomul p, numită 0_p. Funcția f este bine definită pentru că

$\bar{p}_{\mathbf{B}} = \bar{q}_{\mathbf{B}} \Rightarrow p \approx q \Rightarrow 0_p = 0_q$. Se arată că f este un izomorfism boolean:

$$\begin{array}{ccc} P_n & \longrightarrow & P_n(\mathbf{B}) \\ \downarrow & \nearrow & \\ P_n/\approx & & \end{array}$$

Să arătăm acum că dacă *două polinoame* sînt *echivalente pe* $\mathbf{B}$, *adică* $\bar{p}_{\mathbf{B}} = \bar{q}_{\mathbf{B}}$, *atunci funcțiile polinomiale coincid pe orice algebră booleană* B: $\bar{p}_{\mathbf{B}} = \bar{q}_{\mathbf{B}}$.

Demonstrație. Fie $p, q \in P_n$ și $p \approx q$. Fie B o algebră booleană oarecare.

Fie o mulțime M și $\mathbf{B}^M$ algebra tuturor aplicațiilor lui M pe $\mathbf{B}$, avînd funcția caracteristică f_C.

Fie funcția $g : \mathscr{P}(M) \to \mathbf{B}^M$, $C \to f_C$. Se demonstrează că g este un izomorfism. Considerăm acum monomorfismul $B \to \mathscr{P}(M)$ asigurat de teorema de reprezentare (în sens larg). Atunci vom avea și monomorfismul $B \to \mathbf{B}^M$.

Prin definiție $p \approx q \Leftrightarrow \bar{p}_{\mathbf{B}} = \bar{q}_{\mathbf{B}}$, unde $\bar{p}_{\mathbf{B}}(e_1, \ldots, e_n) = \bar{q}_{\mathbf{B}}(e_1, \ldots, e_n)$ pentru toți $e_1, \ldots, e_n \in \mathbf{B}$.

Fie acum funcțiile $f_1, \ldots, f_n$ din $\mathbf{B}^M$ și fie un element $x \in M$. Putem scrie

$$(\bar{p}_{\mathbf{B}^M}(f_1, \ldots, f_n))(x) = \bar{p}_{\mathbf{B}}(f_1(x), \ldots, f_n(x)) =$$

$$= \bar{q}_{\mathbf{B}}(f_1(x), \ldots, f_n(x)) = (\bar{q}_{\mathbf{B}^M}(f_1, \ldots, f_n))(x).$$

Deci $\bar{p}_{\mathbf{B}^M} = \bar{q}_{\mathbf{B}^M}$.

Monomorfismul $B \to \mathbf{B}^M$ permite să conchidem că $\bar{p}_B = \bar{q}_B$.

Un polinom oarecare p este perfect definit de forma sa normală disjunctivă p^N, iar funcția polinomială $\bar{p}_B$ de valorile coeficienților dezvoltării normale disjunctive (v. 3.8). Aceste valori sînt evaluări, deci funcții de forma $\{0, 1\}^n \to \{0, 1\}$. Există o bijecție între $\bar{p}^N_{\mathbf{B}}$ și mulțimea funcțiilor $\{0, 1\}^n \to \{0, 1\}$. Vom avea relațiile

$$|\{\bar{p}_{\mathbf{B}} \mid p \in P_n^N\}| = |\{0, 1\}^{\{0,1\}^n}| = 2^{2^n},$$

$$2^{2^n} = |\{\bar{p}_B \mid p \in N_d\}| \leqslant \{P_n/\approx\} = |P_n(\mathbf{B})| \leqslant |\mathscr{F}_n(\mathbf{B})| = 2^{2^n}.$$

Există deci 2^{2^n} clase de echivalență în P_n,

$$|P_n/\approx| = 2^{2^n}.$$

Rezultă totodată că $P_n(\mathbf{B}) = \mathscr{F}_n(\mathbf{B})$.

În cazul $|B| = 2$, fiecare funcție $\mathbf{B}^n \to \mathbf{B}$ este o funcție polinomială. Spunem că $\mathbf{B}$ este polinomial completă și este singura algebră booleană polinomial completă.

Comentariu. Pentru logician formulele logicii clasice propoziționale F pot fi transcrise ca polinoame booleene P_n, iar tablele de adevăr ca $P_n(B)$:

$$\begin{array}{ccc} F & \longrightarrow & F_n(B) \\ \downarrow & \nearrow & \\ F/\approx & & \end{array}$$

Scanned with OKEN Scanner

Izomorfismul în cazul finit dintre $F/\approx$ și $F_n(B)$ este corespondența biunivocă între clasele de formule echivalente cu tablele de adevăr.

Funcțiile de adevăr sînt în cazul $n = 2$ funcții de forma $\mathbf{B} \times \mathbf{B} \to \mathbf{B}$ așa cum se vede din tabla de adevăr citită algebric,

$\mathbf{B} \times \mathbf{B} \to \mathbf{B}$		
a_1	a_2	$p(a_1, a_2)$
1	1	1
1	0	0
0	1	1
0	0	0

Considerațiile de mai înainte ne duc la

$$|P_2/\approx| = P_2(\mathbf{B}) = |\mathbf{B}^{\mathbf{B}^2}| = 2^4 = 16.$$

Cele 16 elemente reprezentative pentru $P_2/\approx$ sînt:

$$P = \{[a_1 \cup a_1'],\ [a_1 \cup a_2],\ [a_1 \cup a_2'],\ [a_1],\ [a_1' \cup a_2],$$
$$[a_2],\ [(a_1 \cap a_2) \cup (a_1' \cap a_2')],\ [a_1 \cap a_2],\ [a_1' \cup a_2'],$$
$$[(a_1 \cap a_2') \cup (a_1' \cap a_2)],\ [a_2'],\ [a_1 \cap a_2'],\ [a_1'],$$
$$[a_1' \cap a_2],\ [a_1' \cap a_2'],\ [a_1 \cap a_2']\},$$

și care nu sînt altceva decît cei 16 conectori binari posibili din 1.9.

3.10. ALGEBRE BOOLEENE TOPOLOGICE

O *algebră booleană topologică* se numește o algebră booleană B cu o operație I, care asociază fiecărui element $x \in B$ un element $Ix \in B$, astfel ca următoarele condiții să fie îndeplinite pentru toți $x, y \in B$:

1) $I(x \cap y) = Ix \cap Iy$, 3) $I(Ix) = Ix$,

2) $Ix \leqslant x$, 4) $I\,1 = 1$

Operația I se numește *operația interioară*, iar elementul Ix *interiorul lui* x. Elementul $(Ix')'$ se numește *închiderea* lui x și este notat cu Cx. Prin definiție

5) $Cx = (Ix')'$; 5′) $Ix = (Cx')'$.

Se demonstrează că

1′) $C(x \cup y) = Cx \cup Cy$, 3′) $C(Cx) = Cx$,

2′) $x \leqslant Cx$, 4′) $C\,0 = 0$.

Operația C în algebra booleană se numește *operația de închidere*. Dacă C este introdus prin axiomele 1′ — 4′, I poate fi definit prin 5′) și va avea proprietățile 1—4.

Din axiome rezultă că pentru două elemente x, y oarecare din B, $x \leqslant y$ atrage $Ix \leqslant Iy$ și $Cx \leqslant Cy$.

Scanned with OKEN Scanner

Relevanța algebrelor booleene topologice pentru logică provine din faptul că cei doi functori modali fundamentali : $\Box$(certitudinea) și $\Diamond$ (posibilitatea) satisfac condițiile lui I, respectiv C.

Vom enunța cîteva din definițiile și proprietățile care rezultă imediat din axiome și definiții :

1. Un element $x \in B$ este *deschis*, dacă $x = Ix$.

Un element $x \in B$ este *închis*, dacă $x = Cx$.

2. Interiorul unui element este deschis.

Închiderea unui element este închisă.

3. Un element x este deschis dacă și numai dacă complementul său x' este închis.

Un element x este închis dacă și numai dacă complementul său x' este deschis.

4. Produsul $\cap$ unui număr finit de elemente deschise este deschis.

Suma $\cup$ unui număr finit de elemente închise este închisă.

5. Cînd există, suma $\cup$ oricărui număr de elemente deschise este deschisă.

Cînd există, produsul $\cap$ oricărui număr de elemente închise este închis.

5′. Suma $\cup$ unui număr finit de elemente deschise este deschisă.

Produsul $\cap$ unui număr finit de elemente închise este închis.

6. Dacă a este deschis, atunci pentru toți x,

$$a \leqslant x \text{ dacă și numai dacă } a \leqslant Ix.$$

Dacă b este închis, atunci pentru toți x,

$$x \leqslant b \text{ dacă și numai dacă } Cx \leqslant b.$$

6′. Ia este cel mai mare element deschis conținut în a.

Cb este cel mai mic element închis ce conține pe b.

7. Un element a se numește *dens* cînd $Ca = 1$.

8. O clasă C de elemente deschise ale lui B este o *bază* a lui B dacă fiecare element deschis în A este suma unor elemente din C. O clasă B_0 de elemente deschise este o *subbază* a lui B, dacă clasa C compusă din 0 și 1 în toate produsele finite $x_1 \cap x_2 \cap \ldots \cap x_n$, unde $x_1, \ldots, x_n \in B_0$, sînt o bază pentru B. Dacă B_0 este o subbază care conține 0 și 1 și dacă $x \cap y \in B_0$ pentru toți $x, y \in B_0$, atunci B_0 este o bază.

9. Pentru fiecare clasă B_0 de elemente a unei algebre booleene B există o unică operație I și B astfel ca B_0 este o subbază a algebrei booleene topologice B cu operația interioară I.

Demonstrația urmează pas cu pas o teoremă mai generală din topologie.

În algebrele boolene topologice se definește omomorfismul și izomorfismul care păstrează pe lîngă operațiile booleene și operația I :

$$h(Ix) = Ih(x),$$

echivalentă cu $h(Cx) = Ch(x)$.

Asemenea funcții se numesc și *omomorfisme* (și *izomorfisme*) *topologice*.

Algebrele booleene topologice joacă un rol analog în studiul unor logici speciale (intuiționiste, modale) cu algebrele booleene simple pentru ogica propozițională clasică.

Scanned with OKEN Scanner

3.11. ALGEBRELE TEORIILOR PROPOZIȚIONALE

Fie $\mathscr{T} = \{\mathscr{L}, C, \mathscr{A}\}$ o teorie de ordinul 0. Fie F mulțimea formulelor limbajului $\mathscr{L} = \{A, F\}$. Logicile propoziționale de ordinul 0, clasice și neclasice se bazează pe un asemenea limbaj. Este interesant de văzut ce teoreme din teoriile de ordinul 0 (în cazul nostru ce teoreme ale sistemelor logice) determină proprietățile algebrei Lindenbaum ale acestor teorii.

Formăm întîi algebra formulelor teoriei: $\langle F, \wedge, \vee, \rightarrow, \sim\rangle$ și apoi algebra cît (Lindenbaum): $\mathscr{A}(\mathscr{T}) = \langle F/\approx, \wedge, \vee, \rightarrow, \sim\rangle$.

Pornim de la un număr de 13 formule pe care le împărțim în grupuri. Acceptarea treptată a formulelor fiecărui grup ca teoreme are ca efect organizarea treptată a lui $F/\approx$ ca mulțime parțial ordonată, ca latice, ca latice relativ pseudobooleană, ca algebră pseudobooleană și, în sfîrșit, ca algebră booleană.

T0 $\quad \vdash A \rightarrow A$

T1 $\quad \vdash (A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$

T2 $\quad \vdash A \rightarrow A \vee B$

T3 $\quad \vdash B \rightarrow A \vee B$

T4 $\quad \vdash (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$

T5 $\quad \vdash A \wedge B \rightarrow A$

T6 $\quad \vdash A \wedge B \rightarrow B$

T7 $\quad \vdash (C \rightarrow A) \rightarrow ((C \rightarrow B) \rightarrow (C \rightarrow (A \wedge B)))$

T8 $\quad \vdash (A \rightarrow (B \rightarrow C)) \rightarrow ((A \wedge B) \rightarrow C)$

T9 $\quad \vdash ((A \wedge B) \rightarrow C) \rightarrow (A \rightarrow (B \rightarrow C))$

T10 $\quad \vdash (A \wedge \sim A) \rightarrow B$

T11 $\quad \vdash (A \rightarrow (A \wedge \sim A)) \rightarrow A$

T12 $\quad \vdash A \vee \sim A$

Presupunem 1) că Modus Ponens este valabil în $\mathscr{T}$ și
2) $A \leqslant B$ dacă și numai dacă $A \Rightarrow B$, adică

$$\vdash A \rightarrow B.$$

Rezultatele sînt următoarele teoreme:

1. Dacă formulele de forma T0 și T1 sînt teoreme, relația $A \leqslant B$ este relație de cvasiordine în F și o relație de ordine parțială în $F/\approx$.

Demonstrație. Relația este reflexivă: $A \leqslant A$ (T0); este tranzitivă: $A \leqslant B$, $B \leqslant C$ înseamnă că $A \rightarrow B$ și $B \rightarrow C$ sînt teoreme. Din MP aplicat la T1 rezultă că $A \rightarrow C$ este teoremă și $A \leqslant C$.

Relația de echivalență în $\mathscr{T}$ este:

$A \approx B$ dacă $\vdash A \rightarrow B$ și $\vdash B \rightarrow A$, adică $A \leqslant B$ și $B \leqslant A$.

O relație de cvasiordine în F determină o relație de echivalență în F. Se verifică că dacă definim $|x| \leqslant |y|$ dacă și numai dacă $x \leqslant y$ ($x \rightarrow y$ este teoremă), atunci $\leqslant$ este o relație de ordine parțială în $F/\approx$. Mulțimea $F/\approx$ este astfel parțial ordonată.

Scanned with OKEN Scanner

2. Dacă formulele T0 — T7 sînt teoreme în $\mathscr{T}$, atunci $F/\approx$ este o latice, iar $\approx$ este o congruență față de operațiile $\wedge$ și $\vee$.

Fie $|A|$, $|B|$ elemente în $F/\approx$. Cu ajutorul lui T2 și T3, putem scrie

$$|A| \leqslant |A \vee B| \text{ și } |B| \leqslant |A \vee B|.$$

Fie C un alt element al $F/\approx$ astfel încît

$$|A| \leqslant |C| \text{ și } |B| \leqslant |C|.$$

Atunci $\vdash A \to C$ și $\vdash B \to C$. Aplicăm MP la T4 de două ori și obținem $|A \vee B| \leqslant |C|$ sau $|A \vee B| \leqslant |C|$, ceea ce dovedește că $|A \vee B|$ este reuniunea claselor $|A| \vee |B|$. La fel, pentru $|A \wedge B|$. Obținem egalitățile $|A| \vee |B| = |A \vee B|$ și $|A| \wedge |B| = |A \wedge B|$. În consecință, $\mathscr{A}(\mathscr{T})$ este o latice.

3. Dacă $\mathscr{T}$ are drept teoreme formulele T1—T9, atunci $\mathscr{A}(\mathscr{T})$ este o latice relativ pseudocomplementată și $\approx$ este o congruență față de operația $\to$.

Să observăm că T0 este derivabilă din T5, T6, T9 prin MP, deci teoremă în $\mathscr{T}$:

$$(A \to A) \to A \qquad \text{(T5)}$$

$$(((A \wedge A) \to A) \wedge A) \to A \qquad \text{(T5)}$$

$$((((A \wedge A) \to A) \wedge A) \to A) \to (((A \wedge A) \to A) \to (A \to A)) \qquad \text{(T9)}$$

$$((A \wedge A) \to A) \to (A \to A) \qquad \text{MP}$$

$$A \to A \qquad \text{MP}$$

Formulele T0—T7 fiind teoreme, $\mathscr{A}(\mathscr{T})$ este o latice.

Să demonstrăm cerința din definiția pseudocomplementului relativ că pentru orice element C:

$$\text{dacă } |A| \wedge |C| \leqslant |B|, \text{ atunci } |C| \leqslant |A \to B|.$$

Din $|A| \wedge |C| \leqslant |B|$, rezultă că $\vdash C \wedge A \to B$. Din

$$((C \wedge A) \to B) \to (C \to (A \to B)) \qquad \text{(T9)}$$

prin MP rezultă $\vdash C \to (A \to B)$ sau $|C| \leqslant |A \to B|$.

Rămîne să arătăm că $|A| \to |B| = |A \to B|$. Avem nevoie de relația auxiliară

$$|A| \wedge |A \to B| \leqslant |B|.$$

Aceasta se deduce din

$$|A \to B| \to |A \to B| \qquad \text{(T0)}$$

$$((A \to B) \to (A \to B)) \to (((A \to B) \wedge A) \to B) \qquad \text{(T8)}$$

Prin MP obținem $\vdash ((A \to B) \wedge A) \to B$ și $|(A \to B) \wedge A| \leqslant |B|$ sau $|A| \wedge |A \to B| \leqslant |B|$.

Conform definiției $|A| \to |B|$ este cel mai mare element care satisface relația $|A| \wedge |C| \leqslant |B|$, deci

$$|A| \to |B| \geqslant |A \to B|.$$

Scanned with OKEN Scanner

Dar pentru orice element C care satisface această relație, conform demonstrației anterioare

$$|A| \to |B| \leqslant |A \to B|.$$

Am obținut astfel relația căutată.

3′. Un element $|A|$ este elementul unitate al laticei $\mathscr{A}(\mathscr{T})$ dacă și numai dacă A este teoremă în $\mathscr{T}$.

Să arătăm că orice formulă $A \to (B \to A)$ derivă din T5, T6 și T9 prin MP, este deci o teoremă.

Din $A \wedge B \to A$ (T5)

și $((A \wedge B) \to A) \to (A \to (B \to A))$ (T9)

obținem $A \to (B \to A)$ MP

Înseamnă că orice formulă de forma

$$A \to ((A \to A) \to A)$$

este teoremă și prin MP

$$\vdash (A \to A) \to A,$$

pe care o putem scrie

$$|A \to A| \leqslant |A|.$$

Dar laticea relativ pseudocomplementată $\mathscr{A}(\mathscr{T})$ are elementul 1 și $|A| \to |A| = 1$ și astfel $|A \to A| = 1$. În consecință, $|A| = 1$. Reciproc, fie $|A| = 1$. Atunci din teorema $(A \to A) \to A$ și MP rezultă că, $A \to A$ fiind teoremă, A este teoremă.

4. Dacă formulele T1—T11 sînt teoreme, $\mathscr{A}(\mathscr{T})$ este o algebră pseudobooleană. Relația $\approx$ este o congruență față de operația $\wedge$.

Din T10 rezultă că $|A \wedge A^n| \leqslant |B|$ pentru orice B și deci $|A \wedge A^n| = 0$ și $|A| \wedge |A^n| = 0$, ceea ce atrage conform definiției pseudocomplementului într-o latice :

$$|A^n| \leqslant |A|^n$$

Dar $|A|^n = |A| \to 0 = |A| \to |A \wedge A^n| = |A \to (A \wedge A^n)| \leqslant |A^n|$. (La ultima inegalitate a fost folosită T11.) A fost astfel demonstrată proprietatea de congruență

$$|A^n| = |A|^n.$$

5. Dacă într-o teorie formulele T1 — T11 sînt teoreme, $\mathscr{A}(\mathscr{T})$ este o algebră booleană.

Folosim 3′ și T12 pentru a arăta că

$$|A| \vee |A|^n = 1.$$

Deci pseudocomplementul $|A|^n$ este complementul $|A|^\sim$, iar $\mathscr{A}(\mathscr{T})$ este o algebră booleană.

Scanned with OKEN Scanner

Vom rezuma teoremele 1—5 în următorul tablou :

Teoria conține teoreme	Algebra teoriei se organizează ca
T0 — T1	Mulțime parțial ordonată
T0 — T7	Latice
T1 — T9	Latice relativ pseudocomplementată
T1 — T11	Algebră pseudobooleană
T1 — T12	Algebră Boole

Următoarea teoremă se obține ca o consecință a celor precedente. Ea joacă un rol important în demonstrarea consistenței unei teorii.

6. Dacă formulele de forma T1—T9 sînt teoreme într-o teorie $\mathscr{T}$, atunci $\mathscr{T}$ este consistentă dacă și numai dacă laticea $\mathscr{A}(\mathscr{T})$ are cel puțin două elemente (nu este degenerată).

Rezultă din 3′ că $|A| = 1$, dacă și numai dacă A este teoremă. Laticea $\mathscr{A}(\mathscr{T})$ are un element diferit de 1 dacă și numai dacă formula A nu este teoremă (v. 2.7).

7. Demonstrația simplă pe cale algebrică a consistenței unei teorii de ordinul 0 (și a calculului propozițional clasic) este de natură să scoată în relief avantajele abordării algebrice. Pentru demonstrarea proprietăților CP este suficient să luăm în considerare numai evaluările și modelele în algebrele booleene. Proprietățile acestora însă ne îngăduie să restrîngem și mai mult problema. Astfel s-a văzut că echivalența a două polinoame într-o algebră booleană B se reduce la echivalența lor în algebra booleană cu 2 elemente $\mathbf{B}$.

Să recapitulăm și să precizăm cunoștințele dobîndite despre algebra $\mathscr{A}(\mathscr{T})$ care este $F/\approx$ în cazul unei teorii de ordinul 0, în care formulele T1—T11 sînt teoreme, algebră booleană în care se face evaluarea formulelor teoriei logice.

Algebra $\mathscr{A}(\mathscr{T})$ este o algebră booleană. Pentru orice formule A, B din $\mathscr{T}$, elementele lui $\mathscr{A}(\mathscr{T})$ satisfac condițiile :

$$|A| \cup |B| = |A \cup B|$$
$$|A| \cap |B| = |A \cap B|$$
$$|A| \to |B| = |A \to B|$$
$$\sim |A| = |\sim A|$$

Relația $|A| \leqslant |B|$ are loc dacă și numai dacă $A \to B$ este teoremă în $\mathscr{T}$. O formulă este teoremă în $\mathscr{T}$ dacă și numai dacă $|A|$ este elementul 1 din $\mathscr{A}(\mathscr{T})$. O formulă este *nerespinsă* în $\mathscr{T}$ dacă și numai dacă $|A| \neq 0$. Teoria $\mathscr{T}$ este consistentă dacă și numai dacă algebra booleană $\mathscr{A}(\mathscr{T})$ este nedegenerată.

Prin formulă *respinsă* înțelegem o formulă A, astfel ca $\sim A$ să fie teoremă. Altfel A este nerespinsă. Conceptul corespunde pe plan sintactic la nerealizabilitatea și realizabilitatea semantică.

Am considerat evaluarea $v : P \to B$, unde P este mulțimea variabilelor propoziționale și v_0 evaluarea canonică (aplicația naturală de la 3.5), care face să corespundă fiecărei formule clasa formulelor echivalente în $\mathscr{A}(\mathscr{T})$: $A^{v_0} = |A| \in \mathscr{A}(\mathscr{T})$.

O evaluare în algebra booleană B este *model* pentru o formulă A, dacă $A^v = 1$ în algebra booleană B. Iar o formulă este validă într-o algebră

Scanned with OKEN Scanner

booleană B, dacă fiecare evaluare este model pentru A în B. Formula A este validă dacă este validă în fiecare algebră booleană. Dacă A este validă, atunci ea este tautologie, adică și în **B** avem $A^v = 1$.

Cu aceste elemente s-a demonstrat că 1) fiecare formulă derivabilă în CP este validă (prin verificarea tuturor axiomelor T1—T11) și că 2) CP este consistent (altfel am ajunge la o contradicție, folosind pe **B**).

Condiția impusă de definiția validității ca o formulă să fie validă în toate algebrele booleene ar face inoperantă utilizarea acestor algebre, dacă nu ar exista teorema de reprezentare, care permite să ne restrîngem la anumite algebre booleene speciale pentru demonstrațiile proprietăților sistemelor logice.

Teorema de reprezentare este reciproca proprietății oricărei mulțimi de submulțimi a unei mulțimi să se organizeze ca o algebră Boole. Să introducem următoarea noțiune pentru a enunța o formă a teoremei de reprezentare :

Se numește *cîmp de mulțimi* o mulțime $\mathscr{A}$ astfel ca $\bigcup \mathscr{A} \in \mathscr{A}$, $\sim X \in \mathscr{A}$ cînd $X \in \mathscr{A}$ și $X \cup Y \in \mathscr{A}$ cînd $X, Y \in \mathscr{A}$. Rezultă că și $0 \in \mathscr{A}$ și $X \cap Y \in \mathscr{A}$ cînd $X, Y \in \mathscr{A}$. Dacă A este un cîmp de mulțimi, $\langle \mathscr{A}, \cup, \cap, \sim, 0, \bigcup A \rangle$ este o algebră booleană de mulțimi a submulțimilor lui $\bigcup A$. Teorema de reprezentare afirmă că, invers, fiind dată o algebră booleană, ea este izomorfă cu o algebră booleană de mulțimi.

La demonstrarea teoremei de reprezentare (v. 3.7) s-a utilizat o proprietate esențială a algebrei booleene conform căreia fiecare element aparține unui ideal maximal (v. 3.6). Această proprietate se mai numește *teorema idealului prim* (orice ideal prim este maximal într-o algebră booleană).

Următoarea teoremă, care conține teorema de completitudine, scoate în relief rolul jucat de teorema de reprezentare și de teorema idealului prim boolean.

În CP clasic, următoarele enunțuri sînt echivalente :

1) Formula A este derivabilă în CP.
2) Formula A este validă.
3) Formula A este validă în fiecare cîmp de mulțimi B.
4) Avem $A^{v_0} = 1$, unde v_0 este evaluarea canonică.
5) Formula A este tautologie.

Pentru demonstrarea teoremei, implicațiile 1) $\Rightarrow$ 2), 2) $\Rightarrow$ 3), 2) $\Rightarrow$ 4) și 4) $\Rightarrow$ 1) sînt imediate.

Pentru a demonstra că 3) $\Rightarrow$ 5) avem nevoie de teorema reprezentării, conform căreia orice algebră booleană, inclusiv B, poate fi interpretată ca un cîmp de mulțimi.

Pentru a demonstra că 5) $\Rightarrow$ 1) avem nevoie de teorema idealului prim boolean. Într-adevăr presupunem că A nu este derivabilă. Atunci $|A| \neq 1$. Există un ideal maximal Δ în $\mathscr{A}(\mathscr{T})$ astfel ca $|A| \in \Delta$. Fie omomorfismul $\mathscr{A}(\mathscr{T}) \to \mathbf{B}$ care nu este altul decît $\mathscr{A}(\mathscr{T})/\Delta$. Evaluarea $h = hv_0$, unde v_0 este evaluarea canonică în $\mathscr{A}(\mathscr{T})$, are proprietatea $A^v = h(A^{v_0}) = h(|A|) = 0$. Deci A nu este tautologie, contrar ipotezei 5).

Reținînd 1) și 5), regăsim teorema completitudinii CP clasic (demonstrată la cap. 2, o cale mult mai laborioasă), care afirmă că clasa formulelor derivabile în CP coincide cu clasa tautologiilor.

Pentru rolul jucat atît direct în demonstrație, cît și în stabilirea teoremei de reprezentare, teorema idealului prim boolean este considerată ca echivalentul algebric al teoremei de completitudine logică.

Scanned with OKEN Scanner

3.12. NOTĂ ISTORICĂ

Logica propoziţională simbolică are la origine o abordare algebrică. George Boole şi-a propus crearea unei algebre în care variabilele să fie elemente logice (nu propoziţii, ci clase de obiecte), spre deosebire de algebra numerică a vremii sale.

Să luăm exemplul lui Boole. El transcrie algebric fraza :

„Avuţia constă din bunuri limitate, transferabile şi producînd plăcerea sau prevenind suferinţa".

Notînd cu w avuţia, cu s bunuri limitate, cu t transferabile, cu p producînd plăcerea şi cu r prevenind suferinţa, el scrie

$$w = s \cdot t \cdot (p + r \cdot (1 - p)).$$

Observăm că $\cdot$ stă pentru şi, iar $+$, de fapt $\oplus$ este disjuncţia strictă (suma disjunctivă). Dacă fraza ar fi avut sensul disjuncţiei stricte, ar fi scris $p \oplus r$, dar întrucît a produce plăcere nu exclude prevenirea suferinţei, se scrie $p + r \cdot (1 - p)$, $1 - p$ fiind în sensul complementului. Astăzi am fi scris $p \cdot r' + p' \cdot r$. La proprietăţile operatorilor algebrici (comutativitate, asociativitate, distributivitate) Boole adaugă legea $x^2 = x$, din care deduce principiul contradicţiei $x \cdot (1 - x) = 0$. Cu $x^2 = x$ el este la un pas de inelul boolean, dar este încurcat de faptul că $1 + x$ (de fapt $1 \oplus x = 0$) şi -1 nu satisfac ecuaţia sa.

Boole foloseşte formula (corectă) $f(x) = f(1) \cdot x + f(0) \cdot (1 - x)$ pentru a deriva concluzia proprie sistemului său $f(x) = 0 \Rightarrow f(0) \cdot f(1) = 0$. Procedeul său este următorul : formula se mai scrie

$$x = \frac{f(0)}{f(0) - f(1)}, \quad 1 - x = -\frac{f(1)}{f(0) - f(1)},$$

iar $x \cdot (1 - x) = 0$ ne dă $-\dfrac{f(1) \cdot f(0)}{(f(0) - f(1))^2} = 0$, deci $f(1) \cdot f(0) = 0$. Apelînd la formula iniţială

$$f(r) = w - s \cdot t \cdot (p + r \cdot (1 - p)) = 0,$$

obţine $(w - s \cdot t \cdot p)(w - s \cdot t) = 0$ sau, dezvoltînd şi utilizînd $x^2 = x$,

$$s = \frac{w}{t \cdot w + t \cdot p \cdot w - t \cdot \mathrm{p}} = f(w, t, p).$$

Boole aplică apoi formula (reprezentarea sub forma normală disjunctivă) :

$$\begin{aligned} f(x, y, z) = {} & f(1, 1, 1) \cdot x \cdot y \cdot z + f(1, 1, 0) \cdot x \cdot y \cdot (1 - z) + \\ & + f(1, 0, 1) \cdot x \cdot (1 - y) \cdot z + f(0, 1, 1) \cdot (1 - x) \cdot y \cdot z + \\ & + f(1, 0, 0) \cdot x \cdot (1 - y) \cdot (1 - z) + f(0, 1, 0) \cdot (1 - x) \cdot y \cdot (1 - z) + \\ & + f(0, 0, 1) \cdot (1 - x) \cdot (1 - y) \cdot z + f(0, 0, 0) \cdot (1 - x) \cdot (1 - y) \cdot (1 - z) \end{aligned}$$

la funcţia $s + f(w, t, p)$, obţinînd

$$\begin{aligned} s = {} & \frac{1}{1} \cdot w \cdot t \cdot p + \frac{1}{1} \cdot e \cdot t \cdot (1 - p) + \frac{1}{0} \cdot w \cdot (1 - t) \cdot p + \\ & + \frac{0}{-1} \cdot (1 - w) \cdot t \cdot p + \frac{1}{0} \cdot w \cdot (1 - t) \cdot (1 - p) + \end{aligned}$$

Scanned with OKEN Scanner

$$+\frac{0}{0}\cdot(1-w)\cdot t\cdot(1-p)+\frac{0}{0}\cdot(1-w)\cdot(1-t)\cdot p+$$

$$+\frac{0}{0}\cdot(1-w)\cdot(1-t)\cdot(1-p),$$

unde coeficienții sînt calculați conform regulilor algebrice

$$f(1,1,1)=\frac{1}{1\cdot 1+1\cdot 1\cdot 1-1\cdot 1}=\frac{1}{1}.$$

Boole elimină coeficienții de forma $\frac{1}{0}$, $\frac{0}{-1}$, reținînd pe cei cu coeficienții $\frac{0}{0}$ (citit „unii"), adică primii doi termeni

$$w\cdot t\cdot p+w\cdot t\cdot(1-p)=w\cdot t$$

și ultimii doi: $\frac{0}{0}\cdot(1-w)\cdot(1-t)$ și astfel formula sa devine

$$s=w\cdot t+\frac{0}{0}\cdot(1-w)(1-p)+\frac{0}{0}\cdot(1-w)\cdot(1-t),$$

pe care o citește astfel: bunurile limitate sînt fie bunuri transferabile, fie unele bunuri transferabile care nu sînt avuție și nu produc plăcere, fie unele bunuri care nu sînt avuție și nu sînt transferabile.

Scopul de bază a fost realizat: prin intermediul unui calcul algebric, care nu a ținut seama de semnificația simbolurilor, dintr-un enunț valabil s-a obținut un altul.

Urmașii lui Boole, Jevans (1864) și Venn (1881) au adus precizări și clarificări la sistemul său, care a cunoscut diferite axiomatizări pînă la cele de la începutul secolului nostru, apropiate de forma de azi. Sistemul a fost astfel simplificat pînă la algebrele booleene. Dar matematicienii au arătat că sistemul original al lui Boole a fost bazat pe ceea ce astăzi se numește o *algebră a ansamblelor semnate.*

Teorema de reprezentare a algebrelor booleene ca algebre de mulțimi datorită lui M. H. Stone (1936) a dat un mare impuls folosirii algebrelor în logică. Metoda de a trata mulțimea de formule sau clasele de formule echivalente ca algebre abstracte a fost elaborată de Lindenbaum și A. Tarski și aplicată unor clase de logici de către Tarski și McKinsey. Gr. C. Moisil este considerat ca unul din protagoniștii aplicării metodelor algebrice în logică.

3.13. EXERCIȚII ȘI PROBLEME

1. Să se arate că în orice latice L, pentru orice $x, y, z, u \in L$:

$$(x\cap z)\cup(y\cap u)\leqslant(x\cup y)\cap(z\cup u).$$

2. Fie $D(n)$ laticea tuturor divizorilor lui n. Să se construiască diagrama Hasse a laticelor $D(20)$ și $D(21)$ și să se arate că este izomorfă cu laticea $D(420)$.

Scanned with OKEN Scanner

3. Să se demonstreze că o latice L este modulară dacă și numai dacă pentru orice $x, y, z \in L$:

$$(x \cup (y \cap z)) \cap (y \cup z) = (x \cap (y \cup z)) \cup (y \cap z).$$

4. Să se demonstreze că dacă x, y, z sînt elementele unei latice modulare care satisfac $(x \cup y) \cap z = 0$, atunci $x \cap (y \cup z) = x \cap y$.

5. Să se arate că o latice L este ditributivă dacă și numai dacă pentru orice $x, y, z \in L$:

$$(x \cap y) \cup (x \cap z) \cup (y \cap z) = (x \cup y) \cap (x \cup z) \cap (y \cup z).$$

6. Să se arate că într-o latice modulară intervalele $[x, x \cup y]$ și $[x \cap y, y]$ sînt izomorfe.

7. Să se demonstreze că relația

$$[(x \cap y) \cup (x \cap z)] \cap [(x \cap y) \cup (y \cap z)] = x \cap y$$

este adevărată în orice latice pentru toți $x, y, z \in L$.

8. O latice L este metrică dacă există funcția reală $v : L \to \mathbb{R}$, astfel ca pentru toți $x, y \in L$

$$v(x) + v(y) = v(x \cup y) + v(x \cap y),$$

$$y \leqslant x \Rightarrow v(y) \leqslant v(x).$$

Să se arate că o latice metrică este modulară.

9. Să se arate că într-o latice distributivă următoarele două relații sînt echivalente:

$$a \cap b \leqslant x \leqslant a \cup b,$$

$$x = (a \cap x) \cup (b \cap x) \cup (a \cap b).$$

10. Într-o latice distributivă L, $x \cup y = x \cup z$ și $x \cap y = x \cap z$ implică $y = z$.

11. Să se arate că laticea $\{D(18); \text{c.m.m.d.c.}; \text{c.m.m.m.c.}\}$, unde D este mulțimea divizorilor lui 18, nu formează o algebră booleană. Să se deseneze diagrama lui Hasse.

12. Să se arate că laticea $\{D(110); \text{c.m.m.d.c.}; \text{c.m.m.m.c.}\}$, unde D este mulțimea divizorilor lui 110, formează o algebră booleană. Să se deseneze diagrama lui Hasse.

13. Să se demonstreze că laticea divizorilor pozitivi ai lui n este o algebră booleană față de operațiile binare „cel mai mare divizor comun" și „cel mai mic multiplu comun" dacă și numai dacă descompunerea în factori primi a lui n nu conține nici un număr la pătrat.

14. Să se arate că într-o algebră booleană 2^{2^n} există funcții de n variabile.

15. Să se arate că laticele

sînt relativ pseudocomplementate și să se întocmească tabla pentru $\rightarrow$. Să se verifice distributivitatea.

16. Să se arate că într-o algebră booleană B ecuația $a \cup x = 1$ are soluția generală $x = a' \cup u$, unde u este un element arbitrar al lui B.

17. Să se arate că forma generală a soluției ecuației $x \cup y = c$ este de forma $x = c \cap (u \cup v')$, $y = c \cap (u' \cup v)$.

18. Pentru ca ecuația $a \cup x = b$ să aibă o soluție, trebuie ca $a \leqslant b$. Ce formă are soluția generală?

19. O cvasilatice este o mulțime ordonată cu o relație reflexivă și tranzitivă, unde pentru orice pereche de elemente x, y există $\sup(x, y)$, $\inf(x, y)$. Să se arate că cele două operații $\cup$ și $\cap$, ce se definesc în cvasilatice, satisfac legile de asociativitate și absorbție, dar nu și legea comutativității.

20. Să se definească idealele unei latice, în mod analog cu cele ale unei algebre booleene și să se arate că ele formează o latice ordonată de incluziune a mulțimilor.

b) Dacă laticea este finită, orice ideal este ideal principal.

c) O latice distributivă este o algebră booleană dacă și numai dacă fiecare ideal prim este maximal.

21. Care sînt idealele și filtrele laticelor de la problema 15?

22. Să se demonstreze că într-o algebră booleană B mulțimea F este filtru, dacă și numai dacă $F' = \{x' \mid x \in F\}$ este un ideal în B.

Scanned with OKEN Scanner

BIBLIOGRAFIE

1. Birkhoff, G. *Lattice Theory*. Providence, R. I. American Mathematical Society, 1967.
2. Calude, C., Căzănescu, V. E. *Lecții de logică matematică*. Universitatea București, 1984.
3. Carvallo, M. *Monographie des treillis et algèbre de Boole*. Paris, Gauthier-Villars. 1962.
4. Căzănescu, V. E. *Introducere în logica matematică*. București, Universitatea București, 1981.
5. Crawley, P., Dilworth, R. P. *Algebraic Theory of Lattices*. Englewood Cliffs, Prentice Hall, 1973.
6. Grätzer, G. *General Lattice Theory*. Berlin, Akademie Verlag, 1978.
7. Grätzer, G. *Universal Algebra*. New York, Springer Verlag, 1979.
8. Malcev, A. I. *Algebraic Systems*. Berlin, Akademie Verlag, 1973.
9. Monk, J. D. *Mathematical Logic*. New York. Springer Verlag, 1976.
10. Rasiowa, H., Sikorski, R. *The Mathematics of Matemathematics*. Warszawa, P.W.N., 1970.
11. Rudeanu, S. *Boolean Functions and Equations*. North Holland, American Elsevier, 1974.
12. Rudeanu, S. *Latice și algebre booleene*. București, Universitatea București, 1982.
13. Sikorski, R. *Boolean Algebras*. Berlin, Springer Verlag, 1964.
14. Stihi, T. *Elemente de logică matematică cu aplicații*. București, Institutul Politehnic București, 1980.
15. Stoll, R. *Sets, Logic and Axiomatic Theories*. San Francisco, W. H. Freeman Co., 1974.

Scanned with OKEN Scanner

4.

ALGORITMI

4.1 ALGORITMUL LUI WANG

Algoritmul lui Wang stabilește dacă o formulă este tautologie sau nu. Wang folosește secvențele α, β, γ, ... și formulele P, Q, R, ... care satisfac regulile de inferență admisibile (v. 2.9) următoare :

1) $\dfrac{P, \alpha \Rightarrow \beta, \gamma}{\alpha \Rightarrow \beta, \sim P, \gamma}$

2) $\dfrac{\alpha, \beta \Rightarrow \gamma, P}{\alpha, \sim P, \beta \Rightarrow \gamma}$

3) $\dfrac{\alpha \Rightarrow \beta, P, \gamma \,;\, \alpha \Rightarrow \beta, Q, \gamma}{\alpha \Rightarrow \beta, P \wedge Q, \gamma}$

4) $\dfrac{\alpha, P, Q, \beta \Rightarrow \gamma}{\alpha, P \wedge Q, \beta \Rightarrow \gamma}$

5) $\dfrac{\alpha \Rightarrow \beta, P, Q, \gamma}{\alpha \Rightarrow \beta, P \vee Q, \gamma}$

6) $\dfrac{\alpha, P, \beta \Rightarrow \gamma \,;\, \alpha, Q, \beta \Rightarrow \gamma}{\alpha, P \vee Q, \beta \Rightarrow \gamma}$

7) $\dfrac{\alpha, P \Rightarrow \beta, Q, \gamma}{\alpha \Rightarrow \beta, P \to Q, \gamma}$

8) $\dfrac{\alpha, Q, \beta \Rightarrow \gamma \,;\, \alpha, \beta \Rightarrow \gamma, P}{\alpha, P \to Q, \beta \Rightarrow \gamma}$

9) $\dfrac{P, \alpha \Rightarrow \beta, Q, \gamma \,;\, Q, \alpha \Rightarrow \beta, P, \gamma}{\alpha \Rightarrow \beta, P \equiv Q, \gamma}$

10) $\dfrac{P, Q, \alpha, \beta \Rightarrow \gamma \,;\, \alpha, \beta \Rightarrow P, Q, \gamma}{P \equiv Q, \alpha, \beta \Rightarrow \gamma}$

Regulile admisibile sînt variante ale celor demonstrate la 2.9 sau se demonstrează analog.

Procedeul constă în a scrie o formulă sub forma $\alpha \Rightarrow \beta$ și de a o transforma aplicînd regulile de mai sus (de exemplu regulile 1 și 2 admit trecerea unei formule P de cealaltă parte a săgeții, unde devine $\sim P$). Regula fundamentală este următoarea : dacă α, β sînt secvențe de formule, $\alpha \Rightarrow \beta$ este tautologie dacă și numai dacă există o formulă P de ambele părți ale săgeții.

Se demonstrează că metoda este completă : orice tautologie poate fi dovedită ca atare prin acest procedeu.

Practic se aplică succesiv regulile 1) — 10) conectorilor ce apar în formulele de demonstrat, în ordinea apariției lor, pînă la obținerea a două șiruri separate de săgeată. De regulă, dacă conectorul principal este implicația, săgeata îi ia locul. Dacă implicația nu apare, săgeata este pusă în fruntea formulei de demonstrat.

Scanned with OKEN Scanner

Exemple. 1) $\sim(P \vee Q) \rightarrow \sim P$. Scriem formula sub forma $\sim(P \vee Q) \Rightarrow \sim P$.
Aplicăm regula 1 primului conector $\sim$: $\Rightarrow \sim P, P \vee Q$.
Aplicăm regula 2 primului conector $\sim$: $P \Rightarrow P \vee Q$.
Aplicăm regula 5 conectorului $\vee$: $P \Rightarrow P, Q$.
Întrucît P apare în ambele părți, formula dată este o tautologie.
2) Fie $(\sim P \wedge \sim Q) \rightarrow (P \leftrightarrow Q)$. Avem $\sim P \wedge \sim Q \Rightarrow P \equiv Q$ Apoi $\sim P, \sim Q \Rightarrow P \leftrightarrow Q$ și $\sim Q \Rightarrow P \equiv Q, P$. În final $\Rightarrow P \equiv Q, P, Q$.
Formula 8 ne conduce la $P \Rightarrow Q, P, Q$ și $Q \Rightarrow P, P, Q$.
Ambele formule satisfac regula 1.
3) Fie $P \equiv P \wedge (P \vee Q)$. Formula $\Rightarrow P \equiv P \wedge (P \vee Q)$ devine $P \Rightarrow P \wedge [(P \vee Q)]$ și $P \wedge (P \vee Q) \Rightarrow P$.
Prima devine $P \Rightarrow P$ și $P \Rightarrow P \vee Q$ sau $P \Rightarrow P, Q$.
A doua se poate scrie $P, P \vee Q \Rightarrow P$, care devine $P, P \Rightarrow P$ și $P, Q \Rightarrow P$.
Toate cele patru secvențe finale $P \Rightarrow P$, $P \Rightarrow P, Q$, $P, P \Rightarrow P$ și $P, Q \Rightarrow P$ satisfac regula fundamentală.

Observație. Pentru a ușura scrierea programelor au fost folosite simboluri mai simple, inspirate de scrierea prefix, după cum urmează:

$-$ pentru $\Rightarrow$,	D pentru $\vee$,
B pentru $\equiv$,	F pentru $\sim$,
C pentru $\wedge$,	I pentru $\rightarrow$.

Punctele sînt utilizate pentru separarea subformulelor legate de un conector binar.
În această scriere demonstrarea la calculator s-a efectuat pentru cele trei exemple astfel:

1) $FDPQ - FP$

$- FP, DPQ$

$P - DPQ$

$P - P, Q$

2) $- ICFP . FQ . . BPQ$

$CFP . FQ - BPQ$

$FP . FQ - BPQ$

$FQ - BPQ, P$

$- BPQ, P, Q$

$P - Q, P, Q$

$Q - P, P, Q$

3) $- BP . . CP . DPQ$

$P - CP . DPQ$

$P - P$

$P - DPQ$

$P - P, Q$

$CP . DPQ - P$

$P, DPQ - P$

$P, P - P$

$P, Q - P$

Scanned with OKEN Scanner

O formulă scrisă în forma FNC este o conjuncție de clauze, care sînt la rîndul lor disjuncții de literali. Este mai convenabil să privim formula ca o mulțime de clauze, iar clauza ca o mulțime de literali. Astfel formula $S = \bigwedge_{i \leqslant n} C_i$, unde $C = \bigvee_{j \leqslant m} l_j$, va fi tratată drept $S = \{C_i \mid i \leqslant n\}$, numită *mulțimea de clauze*, iar $C = \{l_j \mid j \leqslant m\}$. Faptul că ordinea clauzelor în formulă și ordinea literalilor în clauze nu sînt ținute în evidență nu are consecințe, din cauza comutativității conjuncției și disjuncției.

Definiții. Se numește *clauză vidă* și se notează cu $\square$ o mulțime vidă de literali.

Întrucît nu există nici o evaluare care să facă literalii din clauza vidă adevărați (întrucît ei nu există), clauza vidă este considerată nerealizabilă. Se numește *formulă vidă* o mulțime vidă de clauze, notată cu $\emptyset$. Întrucît orice evaluare face adevărată fiecare din clauzele ce aparțin unei formule vide, ea este considerată realizabilă (tautologie). O formulă S, astfel ca $\square \in S$, este nerealizabilă. Prezența clauzei vide va fi luată drept test al proprietății de nerealizabilitate.

O clauză $C = \{l\}$ constînd dintr-un singur literal se numește *o unitate*. Fie o mulțime de clauze și un literal l; clauzele care conțin pe l se numesc *l-pozitive* ($l \in C$), iar clauzele care conțin pe $\sim l$ se numesc *l-negative* ($\sim l \in C$). O clauză care nu este nici l-pozitivă, nici l-negativă este *l-neutră*.

Clauzele care sînt concomitent l-pozitive și l-negative sînt excluse ca fiind tautologii. Aceasta pentru că dacă literalii opuși l și $\sim l$ aparțin lui C, atunci C este o tautologie. Dacă o tautologie $C \in S$, atunci S este realizabilă dacă și numai dacă $S - \{C\}$ este realizabilă. În mulțimea de clauze a unei formule presupunem că nu sînt tautologii. Notăm S_l^+ mulțimea clauzelor l-pozitive ale formulei S, S_l^- mulțimea clauzelor l-negative și S_l^0 mulțimea clauzelor l-neutre. Pentru fiecare literal l se obține descompunerea $S = S_l^+ \cup S_l^- \cup S_l^0$.

Următoarele mulțimi sînt utile pentru reducerea mulțimii S de clauze la mulțimi mai simple:

$$\mathrm{POZ}_l(S) = S_l^0 \cup \{C - \{l\} \mid C \in S_l^+\}.$$

$$\mathrm{NEG}_l(S) = S_l^0 \cup \{C - \{\sim l\} \mid C \in S_l^-\}.$$

Prima mulțime este astfel formată din clauze care nu conțin literalul l și din clauzele l-pozitive în care acesta a figurat, după eliminarea sa. Analog, a doua mulțime. Ambele nu cuprind, datorită construirii lor, pe l.

Fie S o mulțime de clauze. Cele patru reguli ce poartă numele lui Davis și Putnam sînt următoarele:

1. **Regula tautologiilor.** Eliminăm clauzele din S care sînt tautologii. Mulțimea S' care rezultă este nerealizabilă dacă și numai dacă S este nerealizabilă.

2. **Regula unității.** Dacă există o clauză unitate $l \in S$, obținem pe S', eliminînd toate clauzele ce conțin pe l. Dacă S' este vidă, S este realizabilă. Astfel, producem pe S_2 formată din S' prin eliminarea lui $\sim l$ din S'. Mulțimea S este nerealizabilă dacă și numai dacă S_2 este nerealizabilă.

Observăm că în această notație $S' = S_l^0 \cup S_l^-$, $S_2 = \mathrm{NEG}_l(S) = S_l^0 \cup \{C - \{\sim l\} \mid C \in S_l^-\}$.

Scanned with OKEN Scanner

3. **Regula literalului pur.** Un literal l se numește *pur* dacă și numai dacă $\sim l$ nu apare în nici o clauză. Dacă un literal este pur, eliminăm toate clauzele care îl conțin. Mulțimea S' care rămîne este nerealizabilă, dacă și numai dacă S este nerealizabilă.

În notația introdusă : $S_l^- = \emptyset$, $S' = S_l^0$.

4. **Regula tăieturii.** Eliminăm pe l din clauzele care îl conțin și formăm o nouă mulțime de clauze S_1 din reuniunea clauzelor care l-au conținut pe l și din acelea care au fost l-neutre. Eliminăm pe l din clauzele care îl conțin și formăm pe S_2 din clauzele astfel rezultate și din clauzele l-neutre. Mulțimea S este nerealizabilă dacă și numai dacă $S_1 \cup S_2$ este nerealizabilă.

În notația utilizată $S_1 = \mathrm{POZ}_l(S)$, $S_2 = \mathrm{NEG}_l(S)$.

Începem prin a demonstra ultima regulă, sub forma unei teoreme, din care celelalte două reguli, 2 și 3, decurg drept corolare.

Teoremă. *Fie S o formulă în* FNC *și l un literal. Mulțimea S este realizabilă dacă și numai dacă cel puțin una din perechile de mulțimi $S_1 = \mathrm{POZ}_l(S)$ și $S_2 = \mathrm{NEG}_l(S)$ este realizabilă.*

Fie S realizabilă, deci există o evaluare $S^v = 1$. Atunci $C^v = 1$ pentru toate clauzele $C \in S$. Înseamnă că în fiecare $C \in S$ există un literal m astfel că $m^v = 1$.

C a z u l 1. $l^v = 0$. Deci $l \neq m$. Pentru $C \in S_l^+$, care conține pe m ($m^v = 1$), $(C - \{l\})^v = 1$. Rezultă $\mathrm{POZ}_l(S)^v = 1$.

C a z u l 2. $l^v = 1$. Pentru $C \in S_l^-$, $(C - \{\sim l\})^v = 1$ și $\mathrm{NEG}_l(S)^v = 1$.

Reciproc, fie $\mathrm{POZ}_l(S)^v = 1$ pentru o evaluare v. Definim evaluarea w astfel : $l^w = 0$, $m^w = m^v$ pentru toți literalii $m \neq l$, $\sim l$. Dacă $C \in S_l^0$, atunci $C^w = C^v = 1$. Dacă $C \in S_l^+$, atunci

$$C^w = (C - \{l\})^w = (C - \{l\})^v = 1.$$

Dacă $C \in S_l^-$, atunci $C^w = 1$ pentru că $(\sim l)^w = 1$. Deci $S^w = 1$. Întrucît există o evaluare w astfel ca $S^w = 1$, mulțimea S este realizabilă dacă $\mathrm{POZ}_l(S)$ este realizabilă.

Pornim acum de la ipoteza $\mathrm{NEG}_l(S)^v = 1$ pentru o evaluare v. Definim pe w astfel : $l^w = 1$, $m^w = m^v$ pentru toți $m \neq l$, $\sim l$.

Cînd $C \in S_l^0$, $C^w = C^v = 1$; cînd $C \in S_l^-$, $C^w = (C - \{\sim l\})^w = (C - \{\sim l\})^v = 1$; iar dacă $C \in S_l^+$, $C = 1$ pentru că $l^w = 1$. Concluzia este că și în aceste cazuri $S^w = 1$.

Am obținut astfel regula 4 (a tăieturii) care reduce examinarea realizabilității lui S la aceea a două mulțimi mai ușor de mînuit S_1 și S_2 din care lipsește literalul l. Dacă S_1 este realizabilă, nu mai este nevoie de a cerceta pe S_2. Dar pentru a demonstra că S este nerealizabilă, se cere examinarea ambelor mulțimi S_1 și S_2.

În două cazuri destul de frecvente, studiul lui S_2 este suficient.

Corolarul 1. *Dacă $S_l^- = \emptyset$, mulțimea S este realizabilă dacă și numai dacă $S_2 = \mathrm{NEG}_l(S) = S_l^0$ este realizabilă* (regula 3 a literalului pur).

Presupunem că $S_l^- = \emptyset$. Atunci $\mathrm{NEG}_l(S) = S_l^0 \in \mathrm{POZ}_l(S)$. Pentru orice evaluare pentru care $\mathrm{POZ}_l(S)^v = 1$ rezultă $\mathrm{NEG}_l(S)^v = 1$, deci este suficient să examinăm pe S_2.

Corolarul 2. *Dacă $l \in S$, atunci S este realizabilă dacă și numai dacă $\mathrm{NEG}_l(S)$ este realizabilă* (regula 2 a unității).

Întrucît $\{l\} - \{l\} = \square$, rezultă că $\square \in \mathrm{POZ}_l(S)$ și deci aceasta este nerealizabilă. Examinăm în consecință doar $\mathrm{NEG}_l(S)$.

Scanned with OKEN Scanner

În aplicații cea mai eficientă regulă este aceea a unității, deoarece mulțimea de clauze de examinat este formată doar din clauzele în care figura $\sim l$ și a fost eliminat.

Exemplu. Fie formula $S = (p \vee q \vee r) \wedge (p \vee \sim q) \wedge \sim p \wedge r \wedge u$, care se poate scrie ca o mulțime de clauze

$$S = (p \vee q \vee \sim r,\ p \vee \sim q,\ \sim p,\ r,\ u) \text{ sau}$$

$$\{p, q, \sim r\},\ \{p, \sim q\},\ \{\sim p\},\ \{r\},\ \{u\}.$$

Considerăm literalul unitate $\sim p$. Mulțimea clauzelor de examinat, $\mathrm{NEG}_p(S)$, devine

$$S_2 : \{q, \sim r\},\ \{\sim q\},\ \{r\},\ \{u\} \text{ sau } (q \vee \sim r) \wedge (\sim q) \wedge r \wedge u.$$

Literalul $\sim q$ fiind unitate, scriem pe $S = \mathrm{NEG}(S_2')$:

$$\{\sim r\},\ \{r\},\ \{u\} \text{ sau } \sim r \wedge r \wedge u.$$

Repetăm regula pentru literalul $\sim r$, scriind pe $\mathrm{NEG}(S_2')$: $\{\square\}$, $\{u\}$ sau $\square \wedge u$. În consecință, S este nerealizabilă.

O simplificare mai drastică aduce regula literalului pur l, deoarece reduce formula la partea ei l-neutră.

Exemple. 1) Fie formula $S = (p \vee q) \wedge (p \vee \sim q) \wedge (r \vee q) \wedge (r \vee \sim q)$. Deoarece p este literal pur ($\sim p$ nu figurează în clauzele din S), avem de examinat pe $S_1 : (r \vee q) \wedge (r \vee \sim q)$, unde r este literal pur și aplicăm regula 3 din nou pentru r. Obținem $\emptyset$ și deci formula S este realizabilă.

2) Fie $S = (p \vee q \vee \sim r \vee s) \wedge (\sim p \vee \sim q \vee \sim r \vee s) \wedge (\sim p \vee \sim q \vee \sim r)$. Deoarece $\sim r$ este literal pur, scriem $S_r^0 = \emptyset$ și deci S este realizabilă. Dacă am fi luat pe s drept literal pur: $S_s^0 = \sim p \vee \sim q \vee \sim r$.

Oricare din acești literali sînt puri și sîntem conduși, aplicînd regula 3, din nou la $\emptyset$.

Exemplu pentru aplicarea regulii tăieturii. Fie $S = (p \vee \sim q) \wedge (\sim p \vee q) \wedge (q \vee \sim r) \wedge (\sim q \vee \sim r)$; avem $S_1 = \mathrm{POZ}_p(S) = (\sim q) \wedge (q \vee \sim r) \wedge (\sim q \vee \sim r)$, $S_2 = \mathrm{NEG}_p(S) = q \wedge (q \vee \sim r) \wedge (\sim q \vee \sim r)$.

Aplicînd regula 2 pentru $\sim q$ la S_1 și pentru q la S_2, obținem r și $\sim r$. Din nou se aplică regula 2 pentru $\sim r$ și obținem $\emptyset$ și $\emptyset$, deci S este realizabilă.

Regulile Davis-Putnam permit reducerea problemei, dacă o formulă dată S este realizabilă sau nerealizabilă, la problema mai simplă a examinării unor formule simplificate.

Subsumarea constituie o altă regulă de simplificare. Ea constă în eliminarea clauzelor care conțin alte clauze ale mulțimii.

Teoremă. *Fie clauzele C_1, C_2 astfel ca $C_1 \subset C_2$. Dacă formula S în FNC conține clauzele C_1, C_2 ($C_1, C_2 \in S$), atunci S este realizabilă dacă și numai dacă $S - \{C_2\}$ este realizabilă.*

Dacă S este realizabilă, tot așa este $S - \{C_2\}$. Invers, dacă $(S - \{C_2\})^v = 1$, $C_1 = 1$ și $C_2 = 1$ întrucît $C_2 \subset C_1$. Deci $S^v = 1$.

Regula 2 a unității este o ilustrare a subsumării, pentru că clauza unitară $\{l\}$ produce eliminarea tuturor clauzelor care cuprind pe l, $C \supseteq l$.

Definiții. O mulțime finită de cluaze S se numește *minimal nerealizabilă*, dacă:

1) S este nerealizabilă,

2) pentru toate $R \subset S$, R este realizabilă.

O mulțime finită de clauze S se numește *legată*, dacă pentru orice clauză $C_1 \in S$ care conține pe l ($l \in C_1$) există o clauză $C_2 \in S$, care conține pe $\sim l$ ($\sim l \in C_2$). Fiecărui literal l îi corespunde opusul său în altă clauză a lui S.

Scanned with OKEN Scanner

Proprietatea de realizabilitate minimală exclude subsumarea.

Teoremă. *Dacă S este minimal nerealizabilă, atunci:*

1) *pentru nici o pereche de clauze $C_1, C_2 \in S$ nu are loc relația $C_1 \subset C_2$,*

2) *S este legată.*

Într-adevăr, existența relației $C_1 \subset C_2$ are ca urmare faptul că S este realizabilă împreună cu o submulțime a sa, ceea ce este contrar definiției de minimă nerealizabilitate. Presupunem că S minim realizabilă nu ar fi legată. Înseamnă că există un literal l într-o clauză C, astfel ca $\sim l$ să nu apară în altă clauză și deci $S_l^- = \emptyset$. Prin regula literalului pur, S_l^0 este nerealizabilă. Dar $S_l^0 \subset S$, deci S nu poate fi minim nerealizabilă.

Abordarea algebrică. În cele ce urmează vom aborda algebric procedeul Davis-Putnam, care are ca obiectiv demonstrarea nerealizabilității unei formule FNC. Procedeul Davis-Putnam reduce considerarea realizabilității unei formule la reducerea la o formulă mai simplă, conform regulii $Xl + Yl' \to X + Y$, unde X și Y sînt formule FNC (produse de sume), iar l este un literal. Formula se demonstrează făcînd $l = 1$ și $l = 0$ și obținînd implicațiile tautologice $X \to X + Y$ și $Y \to X + Y$.

Fie formula S în formă FNC. Alegem un literal oarecare l și grupăm clauzele în cele ce conțin pe l, (S_l^+), cele ce conțin pe l', (S_l^-), și cele ce nu conțin pe l, (S_l^0). Atunci

$$S = S_l^+ S_l^- S_l^0,$$

unde

$$S_l^+ = (l + A_1)(l + A_2) \dots (l + A_n),$$

$$S_l^- = (l' + B_1)(l' + B_2) \dots (l' + B_m).$$

Aplicăm formula $(l + A_1)(l + A_2) = l + A_1A_2$. Avem

$$(l + A_1)(l + A_2) = l + lA_1 + lA_2 + A_1A_2 = l + A_1A_2$$

prin legea absorbției.

Pentru n variabile se demonstrează în același fel:

$$(l + A_1)(l + A_2) \dots (l + A_n) = l + A_1 \dots A_n.$$

Aplicăm aceeași formulă produsului de clauze l-negative:

$$(l' + B_1)(l' + B_2) \dots (l' + B_m) = l' + B_1 \dots B_m.$$

Avem o nouă expresie pentru

$$S_l^+ = l + A_1A_2 \dots A_n, \quad S_l^- = l' + B_1B_2 \dots B_m.$$

Folosim acum formula $(l + X)(l' + Y) = lY + l'X$. Avem

$$(l + X)(l' + Y) = ll' + lY + l'X + XY = lY + l'X + XY(l + l') =$$
$$= lY(X + 1) + l'X(Y + 1) = lY + l'X.$$

Introducem în expresia lui S: $S = S_l^+ S_l^- S_l^0 = (lB_1 \dots B_m + l'A_1 \dots A_n)S_l^0$ sau recurgem la formula $Xl + Yl' \to X + Y$:

$$S = lB_1 \dots B_m S_l^0 + l'A_1 \dots A_n S_l^0 \to A_1 \dots A_n S_l^0 + B_1 \dots B_m S_l^0.$$

Din această formulă rezultă că S este realizabilă dacă una din formulele $A_1A_2 \dots A_nS_l^0$ sau $B_1B_2 \dots B_mS_l^0$ este realizabilă și este nerealizabilă dacă ambele sînt nerealizabile.

Formarea produselor $A_1A_2 \dots A_n$ este imediată: produsul clauzelor l-pozitive din care este tăiat l; $B_1B_2 \dots B_m$ este produsul clauzelor l-negative din care este eliminat l'; S_l^0 restul clauzelor ce nu conțin pe l sau l'.

Scanned with OKEN Scanner

Exemplu. Fie $S = (p + q)(q + r)(r + w)(r' + p')(w' + q')(q + r')$ (formula lui Anderson), $S_q^+ = (p + q)(q + r)$, $S_q^- = (w' + q')(q' + r')$, $S_q^0 = (r + w)(r' + p')$, $A_1A_2S_q^0 = pr(r + w)(r' + p') = 0$, $B_1B_2S_q^0 = w'r'(r + w)(r' + p') = 0$. Deci S este nerealizabilă.

Cînd există literal unic, S este de forma

$$S = lS_l^+S_l^-S_l^0.$$

Formula suferă o simplificare majoră, întrucît $lS_l^+ = 1$ conform regulii absorbției $l(l + A) = l$ iterate, iar $lS_l^- = lB_1 \dots B_m$, deci

$$S = lB_1 \dots B_mS_l^0.$$

Exemplu. [4]. $S = (p' + q' + r)(p' + q' + s)(p_1' + q_1' + r_1)(r_1' + s' + s_1)\,pqq_1p_1s_1'$.

Întrucît S are opt literali, tabla de adevăr a acestei formule ar avea $2^8 = 256$ linii. Prin simplificarea succesivă efectuată de regula literalului unic obținem

p: $S = (q' + r)(q' + s)(p_1' + q_1' + r_1)(r_1' + s' + s_1)\,pqq_1p_1s_1'$

q: $S = rs(p_1' + q_1' + r_1)(r_1' + s' + s_1)\,pqq_1p_1s_1'$

q_1: $S = rs(p_1' + r_1)(r_1' + s' + s_1)\,pqq_1p_1s_1'$

p_1: $S = rsr_1(r_1' + s' + s_1)\,pqq_1p_1s_1'$

s_1': $S = rsr_1(r_1' + s')\,pqq_1p_1s_1'$

r_1: $S = rss'pqq_1p_1s' = 0$ $(ss' = 0)$.

Deci $S = 0$ și S este nerealizabilă.

4.3. REZOLUȚIE

Metoda rezoluției este o metodă de reducere a problemei examinării proprietății de nerealizabilitate a unei formule care stă la baza unei largi familii de procedee și algoritmi în demonstrarea automată a teoremelor.

Definiții. Se numește *rezolvent* a două clauze C_1, C_2 în raport cu un literal l,

$$\text{rez}_l(C_1, C_2) = (C_1 - \{l\}) \cup (C_2 - \{\sim l\}),$$

unde $l \in C_1$ și $l \in C_2$.

Teoremă. Fie l un literal și C_1, C_2 clauze, astfel ca $l \in C_1$ și $\sim l \in C_2$. Atunci $C_1, C_2 \models \text{rez}_l(C_1, C_2)$.

Fie v o evaluare pentru care $C_1^v = C_2^v = 1$. Dacă $l^v = 1$, atunci $(C_2 - \{\sim l\})^v = 1$, iar dacă $l^v = 0$, $(C_1 - \{l\})^v = 1$. În ambele cazuri $\text{rez}_l(C_1, C_2)^v = 1$.

Teorema a stabilit că rezolventul a două clauze este o concluzie logică a acestora.

Exemple. Să se determine rezolvenții următoarelor perechi de clauze :

1) $C_1 : P$, $C_2 : \sim P \vee Q$, $\quad \text{rez}_P(C_1, C_2) = Q$,

2) $C_1 : P \vee R$, $C_2 : \sim P \vee Q$, $\quad \text{rez}_P(C_1, C_2) = R \vee Q$,

3) $C_1 : \sim P \vee Q \vee R$, $C_2 : \sim Q \vee S$, $\quad \text{rez}_Q(C_1, C_2) = \sim P \vee R \vee S$,

4) $C_1 : \sim P \vee Q$, $C_2 : \sim P \vee R$, $\quad$ nu are rezolvent.

Scanned with OKEN Scanner

Definiție. Fie S o mulțime de clauze. Se numește *deducție* sau *derivare prin rezoluție* a clauzei C, secvența $C_1, C_2, \ldots, C_n$ astfel ca $C_n = C$ și fiecare C_i $(1 \leqslant i \leqslant n)$ este sau o clauză $C_i \in S$ sau există $j, k < i$ și un literal l astfel ca $C_i = \text{rez}_l(C_j, C_k)$. Secvența C_i este astfel formată fie din clauze ale lui S, fie din rezolvenții clauzelor precedente.

Deducția prin rezoluție a clauzei vide se numește *respingere prin rezoluție* sau o *demonstrație* a lui S.

Se numește *rezolvent al unei mulțimi de clauze* în raport cu un literal reuniunea clauzelor l-neutre cu rezolvenții tuturor perechilor de clauze formate dintr-o clauză aparținînd mulțimii l-pozitive, iar cealaltă clauză mulțimii l-negative:

$$\text{REZ}_l(S) = S_l^0 \cup \{\text{rez}_l(C_1, C_2) \mid C_1 \in S_l^+,\ C_2 \in S_l^-\}.$$

Teoremă. *Fie S o formulă* în FNC *și l un literal. Formula S este realizabilă dacă și numai dacă* $\text{REZ}_l(S)$ *este realizabilă.*

Această teoremă ne oferă posibilitatea de a examina o singură mulțime de clauze (spre deosebire de regula tăieturii, care conduce la două mulțimi de clauze), mai simplă decît S, pentru a determina dacă S este sau nu realizabilă.

Demonstrație. Fie S realizabilă, $S^v = 1$. Dacă $C \in S_l^0$, vom avea și $C \in S$, deci $C^v = 1$. Dacă $C \in \text{rez}_l(C_1, C_2)$, unde $C_1 \in S_l^+$, $C_2 \in S_l^-$, atunci $C_1^v = 1$, $C_2^v = 1$ și, conform teoremei anterioare, $C^v = 1$. Cum pentru toate $C \in \text{REZ}_l(S)$, $C^v = 1$, urmează că $\text{REZ}_l(S)^v = 1$.

Reciproc, fie $\text{REZ}_l(S)^v = 1$. În acest caz, ori $\text{POZ}_l(S)^v = 1$, ori $\text{NEG}_l(S)^v = 1$. Într-adevăr, presupunem că $\text{POZ}_l(S)^v = 0$; întrucît $S_l^0 \subseteq \text{REZ}_l(S)$, avem $(S_l^0)^v = 1$. Deci pentru unele clauze $C_1 \in S_l^+$ trebuie să avem $(C_1 - \{l\})^v = 0$. Totuși pentru toate perechile formate din $C_2 \in S_l^-$ și C_1 trebuie să avem $\text{rez}_l(C_1, C_2)^v = \{(C_1 - \{l\}) \cup (C_2 - \{\sim l\})^v = 1$. Deci pentru toți $C_2 \in S_l^-$ avem $(C_2 - \{\sim l\})^v = 1$, iar $\text{NEG}_l(S)^v = 1$. În condițiile regulii tăieturii, cînd una din mulțimile $\text{POZ}_l(S)$ și $\text{NEG}_l(S)$ este realizabilă, S este realizabilă.

Teoremă. *Fie S o formulă în* FNC. *Dacă există o derivare (prin rezoluție) a clauzei C din S, $S^v = 1$ implică $C^v = 1$.*

Prin această teoremă se generalizează teorema care afirmă că rezolventul a două clauze este o consecință logică a acestor clauze.

Demonstrație. Fie $C_1, C_2, \ldots, C_n = C$ o derivare (prin rezoluție) a clauzei C din S. Ca să dovedim că $C_i^v = 1$ $(1 \leqslant i \leqslant n)$ vom proceda prin inducție. Presupunem că teorema este adevărată pentru $j < i$. Cazul $i = 1$ este trivial. Deci, fie $C_j^v = 1$ pentru toți $j < i$. Dacă $C_i \in S$, atunci $C_i^v = 1$. Altfel, $C_i = \text{rez}_l(C_j, C_k)$, unde $j, k < i$. Prin inducție $C_j^v = C_k^v = 1$ și conform teoremei demonstrate, evaluarea rezolventului lor $C_i^v = 1$.

Teoremă. *O formulă S în* FNC *este nerealizabilă dacă și numai dacă există o respingere (prin rezoluție) a lui S. Clauza vidă nu poate fi consecința logică a unei mulțimi realizabile.*

Presupunem că există o respingere a lui S și că am derivat clauza vidă din S. Dacă $S^v = 1$, conform teoremei anterioare, $\square^v = 1$, ceea ce este absurd. Deci $S^v = 0$ pentru toți v.

Invers, fie S nerealizabilă. Fie $l_1, \ldots, l_n$ lista atomilor care apar în S. Fie $S_0 = S$ și $S_i = \text{REZ}_{l_i}(S_{i-1})$, $i = 1, 2, \ldots, k$.

Fiecare S_i conține atomii l_j pentru care $i < j \leqslant k$. Formula S_k nu va conține atomi și trebuie să fie $\emptyset$ sau clauza vidă $\square$. Știm însă că S_i

Scanned with OKEN Scanner

este nerealizabilă pentru $0 \leqslant i \leqslant k$. De aceea $S_k = \{\square\}$. Fie secvența $C_1, \ldots, C_m$ care constă din toate clauzele lui $S_0 = S$, apoi din toate clauzele lui S_1 și pînă la toate clauzele lui S_k. Aceasta înseamnă că $C_m = \square$. Din definiția lui REZ_i rezultă că $C_1, \ldots, C_m$ este o derivare (prin rezoluție).

Derivarea clauzei vide este esența metodei rezoluției pentru a demonstra că o formulă dată este nerealizabilă.

Exemple. a) Fie S mulțimea de clauze

$$C_1 : \sim P \vee Q, \quad C_2 : \sim Q, \quad C_3 : P.$$

Din C_1 și C_2 prin rezoluție în raport cu Q se obține

$$C_4 : \sim P.$$

Din C_4 și C_3 prin rezoluție în raport cu P rezultă

$$C_5 : \square.$$

Întrucît am derivat clauza vidă, S este nerealizabilă.

b) Fie $S = \{C_1, C_2, C_3, C_4\}$,

$$C_1 : P \vee Q, \quad C_2 : \sim P \vee Q, \quad C_3 : P \vee \sim Q, \quad C_4 : \sim P \vee \sim Q.$$

Să se arate că S este nerealizabilă.

Din C_1 și C_2, rezolventul în raport cu P este $C_5 : Q$;

din C_3 și C_4, rezolventul în raport cu P este $C_6 : \sim Q$; din C_5 și C_6, $C_7 : \square$.

c) Să se arate că $C_1, C_2, C_3 \models C$,

$C_1 : P \to (\sim Q \vee (R \wedge S))$,

$C_2 : P$,

$C_3 : \sim S$, $C : \sim Q$.

Avem de demonstrat că $S = C_1 \wedge C_2 \wedge C_3 \wedge \sim C$ este o contradicție. Tabla de adevăr are 16 linii (2^4). Recurgerea la aceasta nu este convenabilă. Sub forma de clauze, formula S se scrie

$$C_1' : \sim P \vee \sim Q \vee R, \quad C_2' : \sim P \vee \sim Q \vee S, \quad C_3' : P, \quad C_4' : \sim S, \quad C_5' : Q.$$

Dacă folosim regula Davis-Putnam a unității aplicată în raport cu P, examinăm formula

$$S' : (\sim Q \vee R) \wedge (\sim Q \vee S) \wedge (\sim S) \wedge Q,$$

și încă o dată în raport cu Q :

$S'' : R \wedge S \wedge \sim S$ care este nerealizabilă.

Prin rezoluție sînt necesari următorii pași :

$C_6' : \sim Q \vee S$ rezolventul lui C_2', C_3',

$C_7' : S$ rezolventul lui C_6', C_5',

$C_8' : \square$ rezolventul lui C_7', C_1',

S este nerealizabilă și deci $C_1, C_2, C_3 \models C$.

Metoda rezoluției sugerează prin caracterul mecanic al eliminării literalilor opuși din clauzele ce îi conțin următorul algoritm, numit al *saturării de nivel*. Fie S mulțimea de clauze. Se cere să demonstrăm că S este nerealizabilă. Calculăm toți rezolvenții din S, îi adăugăm la mulțimea S, calculăm rezolvenții noii mulțimi ș.a.m.d. pînă cînd obținem clauza

Scanned with OKEN Scanner

vidă. Cu alte cuvinte, generăm șirul

$$S_0 = S$$

$$S_1 = \{\text{rez}(C_1, C_2) \mid C_1, C_2 \in S_0\}$$

$$S_n = \{\text{rez}(C_1, C_2) \mid C_1 \in S_0 \cup S_1 \cup \ldots \cup S_{n-1}, C_2 \in S_{n-1}\}, n = 1, 2, \ldots$$

Aplicarea exhaustivă a rezoluției generează însă clauze inutile, redundante, irelevante și tautologice.

Exemplu. Fie $S = P \vee Q, \sim P \vee Q, P \vee \sim Q, \sim P \vee \sim Q$. Să se arate că S este nerealizabilă.

C_1 : $P \vee Q$,

C_2 : $\sim P \vee Q$,

C_3 : $P \vee \sim Q$,

C_4 : $\sim P \vee \sim Q$,

C_5 : Q, rezolvent (C_1, C_2)

C_6 : P, rezolvent (C_1, C_3)

C_7 : $Q \vee \sim Q$, rezolvent (C_1, C_4)

C_8 : $P \vee \sim P$, rezolvent (C_1, C_4)

C_9 : $Q \vee \sim Q$, rezolvent (C_2, C_3)

C_{10} : $P \vee \sim Q$, rezolvent (C_2, C_3)

C_{11} : $\sim P$, rezolvent (C_2, C_4)

C_{12} : $\sim Q$, rezolvent (C_3, C_4)

C_{13} : $P \vee Q$, rezolvent (C_1, C_7)

C_{14} : $P \vee Q$, rezolvent (C_1, C_8)

C_{15} : $P \vee Q$, rezolvent (C_1, C_9)

C_{16} : $P \vee Q$, rezolvent (C_1, C_{10})

C_{17} : Q, rezolvent (C_1, C_{11})

C_{18} : P, rezolvent (C_1, C_{12})

C_{19} : Q, rezolvent (C_2, C_6)

C_{20} : $\sim P \vee Q$, rezolvent (C_2, C_7)

C_{21} : $\sim P \vee Q$, rezolvent (C_2, C_8)

C_{22} : $\sim P \vee Q$, rezolvent (C_2, C_9)

C_{23} : $\sim P \vee Q$, rezolvent (C_2, C_{10})

C_{24} : $\sim P$, rezolvent (C_2, C_{12})

C_{25} : P, rezolvent (C_3, C_5)

C_{26} : $P \vee \sim Q$, rezolvent (C_3, C_7)

C_{27} : $P \vee \sim Q$, rezolvent (C_3, C_8)

C_{28} : $P \vee \sim Q$, rezolvent (C_3, C_9)

C_{29} : $P \vee \sim Q$, rezolvent (C_3, C_{10})

C_{30} : $\sim Q$, rezolvent (C_3, C_{11})

Scanned with OKEN Scanner

C_{21}: $\sim P$, rezolvent (C_4, C_5)

C_{22}: $\sim Q$, rezolvent (C_4, C_6)

C_{23}: $\sim P \vee \sim Q$, rezolvent (C_4, C_7)

C_{24}: $\sim P \vee \sim Q$, rezolvent (C_4, C_8)

C_{25}: $\sim P \vee \sim Q$, rezolvent (C_4, C_9)

C_{26}: $\sim P \vee \sim Q$, rezolvent (C_4, C_{10})

C_{27}: Q, rezolvent (C_5, C_7)

C_{28}: Q, rezolvent (C_5, C_9)

C_{29}: $\square$ rezolvent (C_5, C_{12}).

Exemplul de mai sus scris pentru o formulă cu patru clauze și cu doi atomi ilustrează fenomenul de „explozie a clauzelor". Toate procedeele rezoluției care au fost elaborate au urmărit o rafinare a acesteia, în sensul eliminării clauzelor inutile și a repetițiilor.

Rezoluție liniară. O strategie de micșorare a numărului rezolvenților este impunerea de restricții. În procesul rezoluției liniare se pornește de la o clauză, al cărei rezolvent cu o altă clauză se calculează, iar în pasul următor se caută rezolventul acestuia cu o altă clauză sau un alt rezolvent, pînă cînd se obține clauza vidă.

Definiție. Fie S o mulțime de clauze și o clauză C_0, numită *clauză vîrf*. O derivare liniară a lui C_n din S cu clauza vîrf C_0 se numește o derivare (prin rezoluție) de forma din fig. 4.1, unde C_{i+1} este rezolventul lui

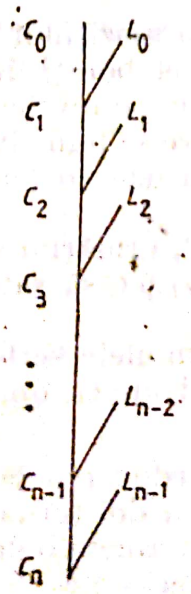


Fig. 4.1

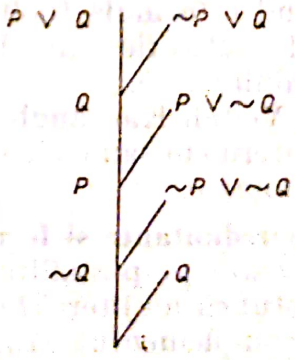


Fig. 4.2

C_i $(i = 0, 1, \ldots, n-1)$, iar L_i aparține lui S sau este un C_j pentru un $j < i$. Clauzele C_i sînt numite *clauze centrale*, iar L_i *clauze laterale*.

Exemplu. În cazul tratat $S = \{P \vee Q,\ \sim P \vee Q,\ P \vee \sim Q,\ \sim P \vee \sim Q\}$ rezoluția liniară se prezintă ca în fig. 4.2.

Abordare algebrică. Fie două clauze $C_1 : l + A$ și $C_2 : l' + B$. Produsul lor este

$$C_1 C_2 = (l + A)(l' + B) = lB + l'A$$

Scanned with OKEN Scanner

și, aplicînd formula $Xl + Yl' \to X + Y$, obținem

$$C_1C_2 \to A + B.$$

Clauza C_1C_2 este realizabilă cînd A sau B este realizabilă și este nerealizabilă, cînd A și B sînt nerealizabile, deci cînd $A+B$ este nerealizabilă. Clauza $A + B$ se numește *rezolventul lui C_1 și C_2 față de literalul l* și se obține prin tăierea literalilor l și l' din cele două clauze și făcînd apoi suma lor.

Exemplu. Fie

$$S = q(p + r)(p' + s)(r' + s)(p' + q' + s')(q' + r' + s').$$

Prin aplicarea regulii literalului unic q:

$$S = q(p + r)(p' + s)(r' + s)(p' + s')(r' + s').$$

Aplicăm rezoluția perechilor $(p + r)(p' + s) \to r + s$, $(p + r)(p' + s') = r + s'$, $(r' + s)(r' + s') = r'$, $S \to q(r + s)(r + s')r'$.

Aplicăm rezoluția perechii $(r + s)(r + s')r'$, $S \to qrr' = 0$, deci S este nerealizabilă.

Observație. Metoda Davis-Putnam și rezoluția reduc examinarea realizabilității unei formule la alta mai simplă, dar între cele două formule nu există echivalență, ci implicație. De aceea metodele nu sînt aplicabile la reducerea la forme minimale.

4.4. METODA VEITCH-KARNAUGH

Posibilitatea scrierii formulelor logicii propozițiilor ca polinoame booleene atrage cu sine folosirea algoritmilor elaborați în algebra booleană pentru manipularea acestor formule, pentru simplificarea și aducerea lor la forme standard. Principalele probleme care se pun sînt minimizarea și descompunerea formulelor, precum și decizia dacă o formulă este realizabilă sau nu.

Dată fiind o formulă (polinom boolean) P, urmărim să găsim o formulă Q astfel ca P să fie echivalent cu Q ($P = Q$) și Q să satisfacă anumite criterii de minimizare.

Metoda Veitch-Karnaugh se referă la formulele scrise în FND (ca sumă de minitermeni sau clauze conjunctive) și are ca obiectiv obținerea de forme minimale.

Forme nereduntante și forme minimale. Reducerea la o formă mai simplă se bazează pe posibilitatea de a combina doi termeni care diferă doar prin faptul că un literal l apare în cei doi termeni o dată ca l și altă dată ca și complementul său, notat cu l': $lm + l'm = m(l + l') = m$.

Exemplu. $x'yz' + x'y'z' = x'z'(y + y') = x'z'$.

Aplicarea repetată a acestei reguli duce la eliminarea literarilor l și obținerea unei formule mai simple.

Exemplu. $P(x, y, z) = x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z$..

Următoarele combinații permit aplicarea regulii:

$$x'yz' + x'y'z' = x'z', \qquad x'y'z' + xy'z' = y'z',$$

$$x'yz' + xyz = yz, \qquad xyz + xy'z = xz.$$

Scanned with OKEN Scanner

Astfel $P(x, y, z) = x'z' + y'z' + yz + xz$.

Această formă este *nereduntantă* întrucît aplicarea regulii de mai sus nu este posibilă. Reducerea nu este unică. Alte combinații sînt posibile:

$$x'yz' + x'y'z' = x'z', \; xy'z' + xy'z = xy', \; x'yz + xyz = yz,$$

ceea ce conduce la

$$P(x, y, z) = x'z' + xy' + yz$$

sau

$$x'yz' + x'yz = x'y, \; x'y'z' + xy'z' = y'z', \; xyz + xy'z = xz,$$

reducere care dă o nouă formă $P(x, y, z) = x'y + y'z' + xz$.

Ambele forme nereduntante astfel obținute au un număr mai mic de termeni și sînt astfel forme minimale. Exemplul a arătat că procedeul poate să conducă la forme neredundante care nu sînt minimale și că formele minimale nu sînt unice.

Metoda hărții (Veitch-Karnaugh). Revenind la tabla de adevăr, codificăm numeric minitermenii prin regula: minitermenii care corespund unei evaluări vor fi numerotați cu numărul zecimal care are drept reprezentare binară evaluarea corespunzătoare minitermenului respectiv. Tablele de adevăr sînt rescrise după această regulă pentru $n = 2, 3, 4$.

Pentru $n = 2$

Cod	x	y	Minitermeni
0	0	0	$x'y'$
1	0	1	$x'y$
2	1	0	xy'
3	1	1	xy

Observație. În scrierea binară 1 este 0 1; 2 este 1 0; 3 este 1 1.

Pentru $n = 3$

Cod	x	y	z	Minitermeni
0	0	0	0	$x'y'z'$
1	0	0	1	$x'y'z$
2	0	1	0	$x'yz'$
3	0	1	1	$x'yz$
4	1	0	0	$xy'z'$
5	1	0	1	$xy'z$
6	1	1	0	xyz'
7	1	1	1	xyz

Pentru $n = 4$

Cod	x	y	z	u	Minitermeni
0	0	0	0	0	$x'y'z'u'$
1	0	0	0	1	$x'y'z'u$
2	0	0	1	0	$x'y'zu'$
3	0	0	1	1	$x'y'zu$
4	0	1	0	0	$x'yz'u'$
5	0	1	0	1	$x'yz'u$
6	0	1	1	0	$x'yzu'$
7	0	1	1	1	$x'yzu$
8	1	0	0	0	$xy'z'u'$
9	1	0	0	1	$xy'z'u$
10	1	0	1	0	$xy'zu'$
11	1	0	1	1	$xy'zu$
12	1	1	0	0	$xyz'u'$
13	1	1	0	1	$xyz'u$
14	1	1	1	0	$xyzu'$
15	1	1	1	1	$xyzu$

Scanned with OKEN Scanner

Cu această convenţie, formulele FND pot fi scrise într-o formă concentrată, ca o sumă urmată de paranteze în care sînt numerele minitermenilor existenţi (evaluarea corespunzătoare care face formula adevărată). În exemplul tratat

$$P(x, y, z) = \Sigma(0, 2, 3, 4, 5, 7).$$

Harta Veitch-Karnaugh (sau diagrama Karnaugh) repre intă o nouă scriere a tablei de adevăr.

Vom trata cazurile $n = 2, 3, 4, 5$.

C a z u l $n = 2$. Pentru a indica locaţia minitermenilor numerotaţi tabla de adevăr pentru $P(x, y)$ se poate aranja astfel :

y \ x	x' 0	x 1
y' 0	0	2
y 1	1	3

Procedura pentru găsirea formei minimale a unei formule date de funcţia sa de adevăr este următoarea :

1) În harta Karnaugh scriem 1 în pătratul ce corespunde evaluării pentru care formula este adevărată, lăsînd goale pătratele în care ea este falsă.

2) Formăm toate combinaţiile de pătrate adiacente în care este scris 1 şi le numim *blocuri*.

3) Unui bloc izolat îi corespunde un termen dat de produsul variabilelor din dreptul său, iar pentru un bloc format din două blocuri adiacente numai variabila liniei sau coloanei respective. Suma acestor termeni constituie forma minimală a formulei date.

Exemple. Formele minimale ale funcţiilor SAU, ŞI, sumă disjunctivă +, NAND şi NOR sînt date de următoarele hărţi Karnaugh :

SAU

y \ x	0	1
0		1
1	1	1

$P = (x, y) = x + y,$

ŞI

y \ x	0	1
0	0	0
1	0	1

$P\,(x, y) = xy,$

+

y \ x	0	1
0	0	1
1	1	0

$P\,(x, y) = xy' + x'y,$

NAND

y \ x	0	1
0	1	1
1	1	

$P\,(x, y) = x' + y'$

NOR

y \ x	0	1
0	1	
1		

$P(x, y) = x'y'.$

Scanned with OKEN Scanner

Cazul $n = 3$. Harta Karnaugh pentru formulele de trei variabile indicînd locarea minitermenilor este următoarea :

z \ xy	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Aplicăm aceleași reguli de înscriere în aceste pătrate a lui 1, în cazul în care tabla de adevăr (și prezența minitermenilor respectivi) indică acest lucru. Formăm apoi blocuri și aplicăm regula 3) modificată :

3') Unui bloc format din două blocuri adiacente termenul corespunzător este produsul variabilelor comune celor două blocuri. Variabila care are valori complementare în cele două blocuri este eliminată.

Exemplu. Fie $P(x, y, z) = x'yz' + xyz' + xyz = \Sigma\ (2, 6, 7)$.

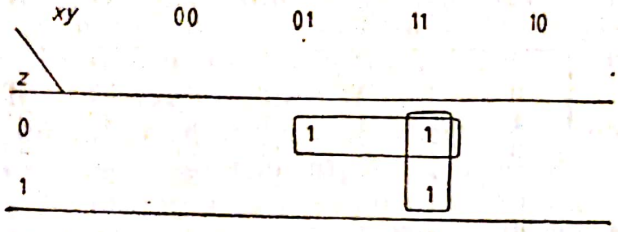


Fig. 4.3

Harta Karnaugh a acestei formule are două blocuri (fig. 4.3). Forma minimală este $P(x, y, z) = yz' + xy$. Primul termen rezultă din blocul $z'(x'y + xy) = z'y$, iar al doilea din blocul $xy\ (z + z') = xy$.

Observație. Aranjarea coloanelor este astfel făcută, încît blocurile adiacente să corespundă la combinații ce diferă printr-o singură variabilă. Astfel, harta este de fapt suprafața unui cilindru, întrucît coloanele 00 și 10 sînt și ele adiacente, elementele lor fiind susceptibile de a forma blocuri.

Cazul $n = 4$. Harta pentru patru variabile este

zu \ xy	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Fie $F(x, y, z, u) = x'yz'u' + x'yz'u + xy'z'u' + xyz'u' + xyz'u + xyzu' + xyzu = \Sigma(4, 5, 8, 12, 13, 14, 15)$.

Harta Karnaugh a acestei formule este dată de fig. 4.4. Cele trei blocuri ale acestei diagrame dau :

blocul coloanei 11 : xy (regula 3'),

blocul pătratelor din linia 00 (11 și 10) : $z'u'x$ (regula 3).

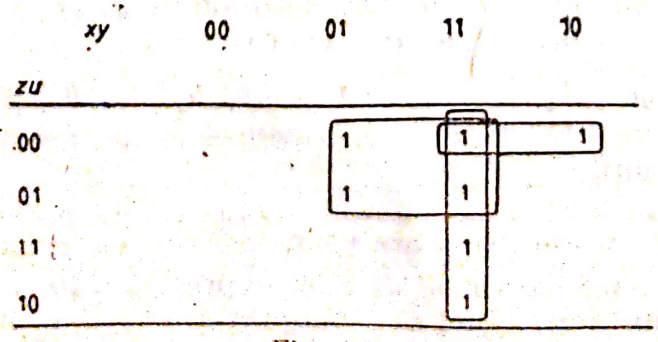


Fig. 4.4.

Cum se scrie termenul corespunzător blocului format de liniile 00, 01 și coloanele 01, 11?

3'') Termenul corespunzător unui bloc pătrat este produsul variabilei, care nu se modifică pe coloana cu variabila care nu se modifică pe linie. În cazul nostru : yz'. Aceasta se poate vedea și din

$$x'yz'u' + x'yz'u + xyz'u' + xyz'u = yz'(x'u' + x'u + xu' + xu) = yz'.$$

Rezultă că forma minimală căutată este

$$F(x, y, z, u) = xy + yz' + xz'u'.$$

Observație. În harta pentru patru variabile prima linie și ultima sînt adiacente, precum și prima coloană și ultima, pătratele lor cu 1 fiind susceptibile de a forma blocuri.

Cazul $n = 5$. Harta Karnaugh pentru cinci variabile este

uv \ xyz	000	001	011	010	110	111	101	100
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

Exemplu. Să se minimizeze formula

$$P(x, y, z, u, v) = \Sigma\,(1, 2, 6, 7, 9, 13, 14, 15, 17, 22, 23, 25, 29, 30, 31).$$

Harta Karnaugh rezultă din fig. 4.5. Forma minimală este

$$P(x, y, z, u, v) = z'u'v + yzv + zu + x'y'uv'.$$

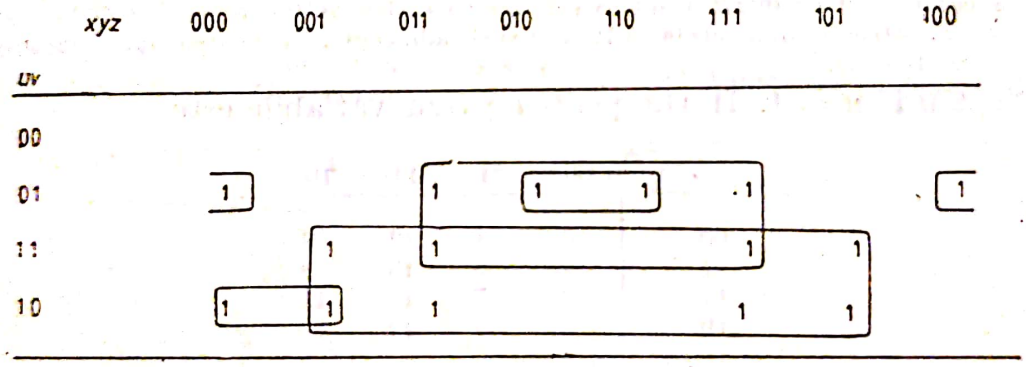


Fig. 4.5.

Forme minimale. Din aplicarea hărților Karnaugh se vede că formele neredundante și cele minimale, care nu coincid, nu sînt necesar unice. Pentru clarificarea formelor minimale se recurge la noțiunea de *implicant prim*.

Definiție. Polinomul p implică polinomul q, $(p \to q)$, cînd pentru toate evaluările $b_1, b_2, \ldots, b_n$ pentru care

$$\overline{p}(b_1, b_2, \ldots, b_n) = 1 \Rightarrow q(b_1, b_2, \ldots, b_n) = 1;$$

p se numește *implicantul lui* q. Se utilizează de asemenea expresia q *acoperă pe* p și notația $q \supseteq p$.

Exemplu. $q = xw + yz$, $p = xy'w$. Avem $p \to q$ (p implică q) pentru că $p = 1$ pentru $x = 1$, $y = 0$, $w = 1$, evaluare pentru care $q = 1$. Echivalent $q \supseteq p$ (q acoperă pe p).

Definiție. *Implicant prim* al unei expresii q este o expresie produs (conectorul $+$ lipsește), astfel că p implică q, dar p nu implică pe q dacă unul din literalii lui p este eliminat. Echivalent : p este un implicant prim

Scanned with OKEN Scanner

al unei expresii q dacă și numai dacă q acoperă pe p, fără ca q să acopere expresiile obținute din p prin eliminarea oricărui literal din p.

Expresiile obținute dintr-o expresie produs prin eliminarea unor literali se numesc *subproduse* ale acesteia. Astfel, r este un subprodus al lui $p = l_1 r l_2$.

Exemple. 1) Fie $q = x'y + xz + y'z'$. Să verificăm că $p = x'y$ este un implicant prim. Într-adevăr, pentru $p = 1$ ($x' = 0$, $y = 1$), $q = 1$ indiferent de valoarea lui z. Deci $p \to q$. Dar subprodusele x' și y ale lui p nu sînt implicanți ai lui q pentru că pentru $x' = 1$ ($x = 0$), $q = y'z'$ și pentru $y = 1$, $q = x' + xz$.

2) Fie expresia $q = x_1x_2x_3 + x_1x_2'x_3 + x_1'x_2'x_3'$. Observăm că $p = x_1x_3$ este un implicant al lui q, deoarece pentru evaluarea $(1, x_2, 1)$ pentru care $p = 1$ și $q = 1$ (indiferent de valoarea lui x_2). Dar $x_1 \to q$ întrucît pentru $x_1 = 1$, $q = x_3$; la fel x_3 nu implică q întrucît pentru $x_3 = 1$, $q = x_1$.

Orice polinom poate fi adus la o formă neredundantă, prin aplicarea regulii $lm + l'm = m$, avînd drept consecință reducerea numărului de literali. Forma neredundantă este obținută cînd aplicarea regulii (și reducerea de literali) nu mai este posibilă.

Orice formă redundantă a lui q este o sumă de implicanți primi ai lui q. Într-adevăr, dacă aceasta ar conține un termen produs, care nu este implicant prim al lui q, ar însemna că este posibilă înlocuirea acestuia printr-un produs cu un număr mai mic de literali, ceea ce ar contrazice ipoteza inițială.

Definiție. Un implicant prim p al unei expresii q este *esențial* dacă acoperă cel puțin un minitermen al lui q, neacoperit de un alt implicant prim.

Forma minimală este o sumă de implicanți primi esențiali.

Următorul exemplu elucidează legătura între formele redundante (sume de implicanți primi) și formele minimale (sume de implicanți primi esențiali) în folosirea hărții Karnaugh.

Exemplu. Fie $f = x'y'z'u' + x'yz'u' + x'yzu + xy'z'u' + xy'z'u + xyz'u + xyzu = \Sigma(0, 4, 5, 7, 8, 9, 13, 15)$.

Harta Karnaugh constă în introducerea lui 1 în căsuțele ce corespund celor opt minitermeni:

zu \ xy	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

zu \ xy	00	01	11	10
00	1	1		1
01		1	1	1
11		1	1	
10				

Prin aplicarea regulii $lm + l'm = m$ obținem

$$f = x'z'u' + y'z'u' + z'yz' + x'yu + xy'z' + xz'u + yzu + xyu =$$
$$= yu + x'z'u' + xy'z' + y'z'u' + x'yz' + xz'u.$$

Mulțimea celor șase implicanți primi este

$$P = \{yu, x'z'u', xy'z', y'z'u', x'yz', xz'u\}.$$

Într-adevăr, cînd valoarea unui implicant prim este 1, unul din termenii lui f este 1 și deci întreaga sumă este 1.

Fiecare implicant prim corespunde unui bloc din harta Karnaugh. Astfel, primul implicant prim yu este rezultatul formării blocului din centru, iar ultimul implicant prim este blocul dreptunghiular alăturat din fig. 4.6. Suma acestor minitermeni nu reprezintă pe f pentru că au rămas minitermeni neacoperiți. Pentru a găsi o formă neredundantă a lui f nu este necesar să formăm toate cele șase blocuri posibile, ci atîtea blocuri cîte sînt necesare să acoperim toți minitermenii. Combinația următoare satisface această cerință (fig. 4.7). Forma neredundantă corespunzătoare, avînd patru termeni (blocuri) este

$$f = y'z'u' + x'yz' + xz'u + yu.$$

Deși mai scurtă decît forma redundantă, aceasta nu este minimală, doi minitermeni fiind acoperiți de cîte două blocuri.

Scanned with OKEN Scanner

Este posibil în cazul discutat să formăm o configurație de blocuri reprezentînd implicanți primi esențiali (fig. 4.8). Această hartă ne dă

$$f = x'z'u' + xy'z' + yu.$$

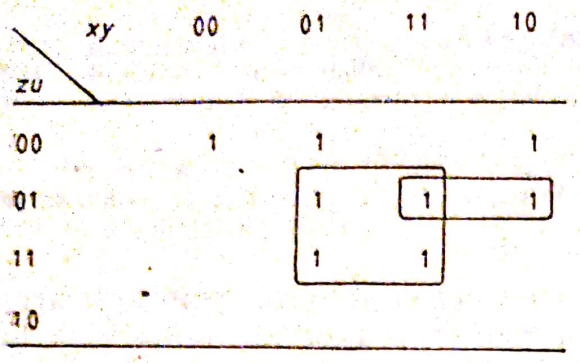


Fig. 4.6

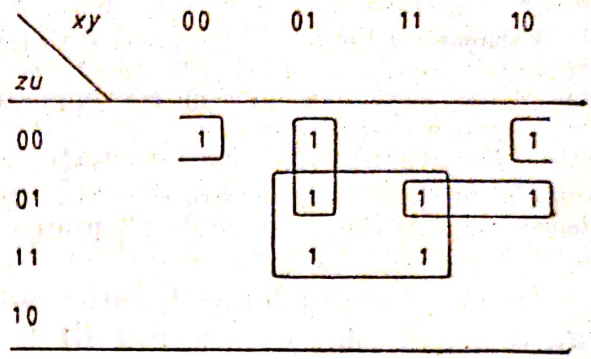


Fig. 4.7

Sîntem în prezența unei forme minimale unice, sumă de implicanți primi esențiali.

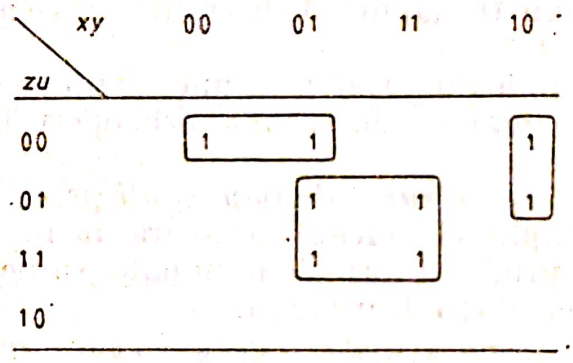


Fig. 4.8

Observații. 1) Întrucît fiecare minitermen acoperit de un implicant prim neesențial este acoperit de cel puțin doi implicanți primi, orice implicant prim neesențial este acoperit de suma unor implicanți primi. Astfel, implicantul prim $x'yz'$ din fig. 4.8 este acoperit de implicanții primi yu și $x'z'u'$. Un implicant prim esențial nu este acoperit de o asemenea sumă.

2) În timp ce implicanții primi esențiali sînt conținuți în orice expresie neredundantă, un implicant prim oarecare acoperit de suma de implicanți primi nu trebuie să fie cuprins într-o expresie neredundantă.

De aici rezultă următoarele reguli pentru obținerea unei forme minimale cu ajutorul hărților Karnaugh :

1. Determinăm pe toți implicanții primi esențiali și îi includem în forma minimală.

2. Îndepărtăm din lista implicanților primi pe cei ce sînt acoperiți de implicanți primi esențiali.

3. Dacă mulțimea de la punctul 1 acoperă toți minitermenii, ea constituie expresia unică minimală. În caz contrar, adăugăm alți implicanți primi, astfel ca f să fie acoperită complet și numărul total de implicanți primi adăugați să fie minimal.

Punctul 3 presupune inspecția hărții, ceea ce este dificil în cazul unui număr de variabile. Metoda următoare oferă o soluție mai sistematică.

Scanned with OKEN Scanner

4.5. ALGORITMUL QUINE-McCLUSKEY

Algoritmul este elaborat pentru determinarea unei forme minimale şi constă în : a) determinarea implicanţilor primi şi b) selectarea implicanţilor primi esenţiali.

Vom enunţa paşii algoritmului concomitent cu ilustrarea aplicării la polinomul $p = xy'z + x'yz' + xyz' + xyz$.

P a s u l 1. Fie polinomul p dat sub forma FND. Reprezentăm toţi minitermenii (expresii produs) prin evaluarea la care ei corespund în tabla de adevăr. Astfel, minitermenii devin secvenţe de 1 şi 0, unde x este înlocuit de 1 şi x' de 0 :

$xy'z$	1 0 1
$x'yz'$	0 1 0
xyz'	1 1 0
xyz	1 1 1

P a s u l 2. Aşezăm reprezentarea minitermenilor în clase, fiecare cuprinzînd minitermenii cu acelaşi număr de 1, iar clasele le ordonăm după numărul crescător al simbolurilor 1 :

1	$x'yz'$	0 1 0
2	xyz'	1 1 0
	$xy'z$	1 0 1
3		
4	xyz	1 1 1

P a s u l 3. Comparăm fiecare din expresiile conţinînd r simboluri 1 cu fiecare din expresiile care conţin $r + 1$ simboluri 1. Dacă acestea diferă într-o singură poziţie (0 şi 1), le adunăm conform regulii eliminării unui literal : $lm + l'm = m$.

În locul literalului eliminat scriem —. Repetăm pînă cînd regula nu se mai poate aplica :

1 , 2	— 1 0	yz'	A
2 , 4	1 1 —	xy	B
3 , 4	1 — 1	xz	C

Am obţinut o listă a implicanţilor primi pe care îi notăm A, B, C, ceea ce aduce pe p la forma

$$p = A + B + C = yz' + xy + xz.$$

Observăm că în timp ce din 1 şi 2 se obţine — 1 0, ceea ce corespunde la $x'yz' + xyz' = (x' + x)yz' = yz'$, din 1 şi 3 nu se obţine nimic, deoarece 0 1 0 şi 1 0 1 nu sînt identice în afara unei singure poziţii, fapt ce corespunde la imposibilitatea reducerii unui literal prin suma $x'yz' + xy'z$.

P a s u l 4. Formăm tabla implicanţilor primi, avîndu-i pe aceştia drept linii, coloanele fiind minitermenii funcţiei date. Examinăm pe

rînd dacă un implicant prim din linie este subprodus al minitermenilor din coloană și în caz pozitiv scriem semnul $\times$ în tablă, în locul de intersecție a liniei și coloanei respective :

				(1)	(2)	(3)	(4)
				0	1	1	1
				1	1	0	1
				0	0	1	1
—	1	0	A	$\otimes$	$\times$		
1	1	—	B		$\times$		$\times$
1	—	1	C			$\otimes$	$\times$

Observăm că A acoperă (1) și (2), întrucît combinația — 1 0 se regăsește în poziția 2 și 3 a minitermenilor (1) și (2), ceea ce corespunde la faptul că yz' este subprodus al lui $x'yz'$ și al lui xyz' .

Tabla servește la identificarea implicanților primi esențiali, care sînt dați de definiția acestora : acoperă cel puțin un minitermen neacoperit de un implicant prim, deci sînt singuri pe coloană. Încercuim semnele respective.

Am obținut implicanții primi esențiali, a căror sumă formează *nucleul* $N = A + C$.

Sînt două cazuri :

a) Implicanții primi esențiali acoperă pe toți minitermenii. În acest caz funcția este egală cu nucleul (suma implicanților primi esențiali) și forma obținută este unic minimală. În exemplul studiat, A acoperă pe (1) dar și pe (2), iar C acoperă pe (3), dar și pe (4). Deci $f = A + C = yz' + xz$.

b) Implicanții primi esențiali nu acoperă toți minitermenii. Formăm nucleul și construim a doua tablă, avînd ca linii pe implicanții primi necuprinși în nucleu și drept coloane minitermenii neacoperiți de nucleu. În această tablă căutăm cel mai mic număr de implicanți primi care acoperă minitermenii și îi adăugăm la nucleu pentru a obține o formă minimă. Sîntem conduși la mai multe soluții, ce corespund la forme minimale alternative, așa cum rezultă din exemplul următor, pentru care aplicăm doar regulile, fără comentarii.

Exemplu. Fie $f = x'y'z'w' + x'y'zw' + x'yz'w + x'yzw' + x'yzw + xy'z'w' + xy'wz + xyz'w + xyzw + xyzw'$.

Pasul 1. Reprezentare

0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
1	1	1	0

Pasul 2. Ordonare

(1)	0	0	0	0
(2)	0	0	1	0
(3)	1	0	0	0
(4)	0	1	0	1
(5)	0	1	1	0
(6)	0	1	1	1
(7)	1	0	1	1
(8)	1	1	0	1
(9)	1	1	1	0
(10)	1	1	1	1

Scanned with OKEN Scanner

Pasul 3. Adunare

(1), (2)	0	0	—	0	A
(1), (3)	—	0	0	0	B
(2), (5)	0	—	1	0	C
(4), (6)	0	1	—	1	
(4), (8)	—	1	0	1	
(5), (6)	0	1	1	—	
(5), (9)	—	1	1	0	

(6), (10)	—	1	1	1	
(7), (10)	1	—	1	1	D
(8), (10)	1	1	—	1	
(9), (10)	1	1	1	—	
(4), (6) — (7), (10)			—	1	— 1 E
(4), (8) — (6), (10)			—	1	— 1
(5), (6) — (9), (10)			—	1	1 — F
(5), (9) — (6), (10)			—	1	1 —

Pasul 4. Tabla implicanților primi

					(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
					0	0	1	0	0	0	1	1	1	1
					0	0	0	1	1	1	0	1	1	1
					0	1	0	0	1	1	1	0	1	1
					0	0	0	1	0	1	1	1	0	1
0	0	—	0	A	×	×								
—	0	0	0	B	×		⊗							
0	—	1	0	C		×			×					
1	—	1	1	D							⊗			×
—	1	—	1	E				⊗		×		×		×
—	1	1	—	F					×	×			⊗	×

Implicanții primi esențiali sînt B, D, E, F, iar nucleul este $B + D + E + F$.
Tabla minitermenilor neacoperiți este

					(2)
					0
					0
					0
					0
0	0	—	0	A	×
0	—	1	0	C	×

Forma minimală este $B + D + E + F + A$ sau $B + D + E + F + C$. Cele două forme minimale sînt

$$f = y'z'w' + xzw + yw + yz + x'y'w',$$

$$f = y'z'w' + xzw + yw + yz + x'zw'.$$

4.6. SIMPLIFICARE ALGEBRICĂ

Putem considera un procedeu de aplicare a formulelor utile din algebra booleană pentru a aduce o formulă dată la o formă mai simplă, eventual minimală, dacă ea este realizabilă. Pentru aceasta vom ține seama de

Scanned with OKEN Scanner

următoarele reguli:

1) *Scriere duală (regula D).* Dacă o formulă este sub forma FND, este convenabil să recurgem la scrierea duală: $+$ în loc de $\cdot$ și $\cdot$ în loc de $\vee$ și nu cum am procedat pînă acum: $+$ a corespuns disjuncției $\vee$ și $\cdot$ conjuncției $\wedge$.

Acest procedeu nu coincide cu aducerea formulei la forma FND, care presupune reîntoarcerea la tabla de adevăr.

Exemplu. Fie $f = xyz + x'yz + xy'z'$. Prin recurgerea la tabla de adevăr vom scrie FNC sub forma unui produs al celor cinci maxitermeni care corespund evaluărilor pentru care $f=0$;

$$f = (x' + y' + z)(x' + y + z')(x + y' + z)(x + y + z')(x + y + z).$$

În timp ce simplificarea lui f se poate face prin scrierea sub forma duală $f = (x + y + z)(x' + y + z)(x + y' + z')$, efectuarea simplificării și revenirea la scrierea curentă se face înlocuind $+$ cu $\cdot$ și $\cdot$ cu $+$.

În scrierea curentă eliminarea conectorului implicație se face prin formula $(a \to b) = a' + b$. În scrierea duală formula devine $(a \to b) = a'b$.

Formulele conținînd $\to$ pot fi simplificate mai ușor prin scrierea duală.

Exemplu. Fie formula lui Leibniz

$$f = (a \to b)(c \to d) \to ((a + c) \to (b + d)).$$

Se observă că aceasta este o tautologie; în calcul boolean înseamnă să o aducem la o formă echivalentă $f = 1$. În scrierea duală

$$f = (a'b + c'd) \to (ac)'bd = (a'b + c'd)'(ac)'bd =$$

$$= (a + b')(c + d')(a' + c')bd = ac(a' + c')bd = 0,$$

folosind regula literalului unic. Întrucît $f^* = 0$, $f = 1$, deci este o tautologie.

2) Regula R1 este regula literalului unic:

$$R1: \; l(l + A) = A,$$

care corespunde *legii absorbției*. Întrucît putem recurge la oricare din lateralii formulei, este convenabil să începem cu literalul cu numărul cel mai mare de apariții.

3) Regula $R2$ este regula eliminării literalului

$$R2: \; (l + A)(l' + A) = A,$$

care stă la baza procedeelor Karnaugh și Quine-McCluskey.

Exemplu. Fie

$$f = xy'z + x'yz' + xyz' + xyz,$$

pe care am adus-o la forma cea mai simplă $f = yz' + xz$, prin algoritmul Quine-McCluskey. Scriem dualul

$$f^* = (x + y' + z)(x' + y + z')(x + y + z')(x + y + z)$$

și aplicăm formula $(l + A)(l' + A) = A$ factorilor 1, 4 și 2, 3, obținînd $f^* = (x + z)(y + z')$. Revenind la scrierea ordinară, avem $f = xz + yz'$.

4) Regula $R3$ este

$$R3: \; (l + A)(l + B) = l + AB.$$

5) Regula $R4$ este

$$R4: \; (l + A)(l' + B) = lB + l'A.$$

Scanned with OKEN Scanner

Spre deosebire de algoritmii Davis-Putnam și al rezoluției vom păstra această formă în loc de a ne folosi de implicația $lB + l'A \to A + B$.

6) Întrucît regulile $R2$ și $R3$ conduc la factori care nu sînt clauze disjunctive (nume de literali), este util să vedem dacă produsul unor asemenea factori va păstra forma factorilor. Răspunsul este pozitiv și considerăm regula

$$R5\text{: } (Al + Bl')(Cl + Dl') = ACl + BDl'.$$

Demonstrația este imediată.

Exemplu. Considerăm din nou formula nerealizabilă Anderson, examinată cu procedeul Davis-Putnam:

$$S = (p + q)(q + r)(r + w)(r' + p')(w' + q')(q' + r').$$

Aplicăm $R3$: $(l + A)(l + B) = l + AB$:

$$(p + q)(q + r) = q + pr,\ (w' + q')(q' + r') = q' + r'w'.$$

Aplicăm $R4$: $(l + A)(l' + B) = lB + l'A$:

$$(q + pr)(q' + r'w') = qr'w' + q'pr,\ (r + w)(r' + p') = rp' + r'w.$$

Aplicăm $R5$: $(Al + Bl')(Cl + Dl') = ACl + BDl'$:

$$(rpq' + r'qw')(rp' + r'w) = rpq'p' + r'qw'w = 0.$$

4.7. SCRIEREA GALOIS

În logica clasică a propozițiilor s-a arătat o predilecție pentru folosirea următoarelor mulțimi de conectori adecvați (capabili să exprime orice formulă logică): $\{\sim, \wedge, \vee\}$, cu ajutorul cărora sînt scrise majoritatea formulelor și algoritmilor, și $\{\sim, \to\}$ adoptată în sistemele formale axate pe demonstrație și deducție, deși conectorii lui Sheffer și Nicod au proprietatea de a forma fiecare cîte o mulțime adecvată.

Dacă adoptăm perechea de conectori $(\oplus, \wedge)$, constatăm că aceasta formează o mulțime adecvată. Este suficient să arătăm că disjuncția și negația se exprimă în funcție de acești conectori. Introducem suma disjunctivă prin tabla sa de adevăr și o comparăm cu tabla de adevăr a disjuncției simple:

a	b	$a \oplus b$	$a \wedge b$
1	1	0	1
1	0	1	1
0	1	1	1
0	0	0	0

Se vede că

$$a \oplus b = (a \wedge (\sim b)) \vee ((\sim a) \wedge b) \text{ și } a \vee b = a \oplus b \oplus (a \wedge b).$$

Făcînd $b = 1$ în această formulă, se obține $1 \oplus a = a$.

Scanned with OKEN Scanner

Vom utiliza de acum înainte scrierea algebrică : $+$ pentru $\vee$, $\cdot$ pentru $\wedge$, $'$ pentru $\sim$. Trecerea de la un sistem la altul se face prin :

$$a \oplus b = ab' + a'b, \qquad a + b = a \oplus b \oplus ab,$$
$$a \cdot b = ab, \qquad ab = a \cdot b,$$
$$1 \oplus a = a', \qquad a' = 1 \oplus a.$$

Aceasta este tocmai transformarea cerută de transformarea unei algebre booleene într-un inel boolean idempotent ($x^2 = x$), comutativ și de caracteristică $2 (x + x = 0)$.

Scriind formulele logicii propozițiilor cu conectorii $\oplus$, $\cdot$ ne situăm în cadrul unui inel de caracteristică $r = 2$. Reamintim proprietățile claselor de resturi care ilustrează acest inel.

Două numere sînt în aceeași clasă modulo 2 cînd $a - b = 2K$ și scriem $a \equiv b \pmod 2$. Egalitatea (mod 2) are proprietățile reflexivității, simetriei și tranzitivității. Cele două clase de resturi (mod 2) sînt C_0 și C_1 (restul împărțirii la 2 fiind 0 și 1), iar operațiile dintre ele sînt date de tabla de adevăr a conectorilor $+$ și $\cdot$.

$\oplus$	C_0	C_1
C_0	C_0	C_1
C_1	C_1	C_0

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	C_0	C_1
C_0	C_0	C_0
C_1	C_0	C_1

$\cdot$	0	1
0	0	0
1	0	1

Acest inel modulo 2 este cazul cel mai simplu al algebrei cîmpurilor Galois, GF(2). Nu este dificil de a extinde calculul la un inel de caracteristică r, pentru r un număr prim, deși nu mai sîntem conduși la o algebră consistentă în cazul cînd r este neprim. Cazul cînd $r = p$, unde p este un număr prim, este însă o excepție. Scrierea logică cu $\oplus$ și $\cdot$ se numește *scrierea Galois*.

Proprietățile conectorilor $\oplus$ și $\cdot$. În afara proprietăților de închidere ale lui $\oplus$ și $\cdot$, ($a \oplus b$ și $a \cdot b$ conduc la elemente ale inelului) se mențin proprietățile de comutativitate, asociativitate, distributivitate stabilite pentru $+$ și $\cdot$.

Comutativitatea : $a \oplus b = b \oplus a$, $\quad ab = ba$.
Asociativitatea : $a \oplus (b \oplus c) = (a \oplus b) \oplus c$, $\quad a(bc) = (ab)c$.
Distributivitatea : $a(b \oplus c) = ab \oplus ac$.

Observație. Distributivitatea adunării față de înmulțire există în algebra booleană :

$$a + bc = (a + b)(a + c)$$

dar nu este valabilă în GF(2) :

$$a \oplus bc \neq (a \oplus b)(a \oplus c).$$

Demonstrație. Avem $a \oplus bc = a(bc)' + a'bc = a(b' + c') + a'bc = ab' + ac' + a'bc$, $(a \oplus b)(a \oplus c) = (ab' + a'b)(ac' + a'c) \neq ab'c' + a'bc$. Dar

$$ab' + ac' = ab'(c + c') + ac'(b + b') = ab'c + ab'c' + ac'b + ac'b' \neq ab'c'.$$

Scanned with OKEN Scanner

Elementul neutru : $a \oplus 0 = a$, $a \cdot 1 = a$.

Inversul : $a \oplus a = 0$.

În GF(2), fiecare element este propriul invers aditiv. Îl vom nota $\bar{a}$ sau $-a$.

Datorită definiției avem $1 \oplus a = \bar{a}$.

Comportamentul lui $\bar{a}$ față de sumă și disjuncție booleană este acela al complementului :

$$a + \bar{a} = a + (1 \oplus a) = a + a' = 1, \quad a \cdot \bar{a} = a(1 \oplus a) = a \oplus a = a \cdot a' = 0.$$

Deci am putea păstra notația $\bar{a} = a' = \sim a$.

Se verifică : $a \oplus \bar{a} = 1$.

Deosebirea față de algebra booleană este netă :

$$a + a = a \text{ și } 1 + a = 1.$$

Din $a \oplus a = 0$, rezultă imediat următoarele relații :

Dacă $a \oplus b = c$, atunci $a = b \oplus c$, $b = a \oplus c$ și $a \oplus b \oplus c = 0$.

Formulele lui De Morgan nu se mențin :

$$a \oplus b \neq \bar{a} \cdot \bar{b} \text{ și } \overline{ab} \neq \bar{a} \oplus \bar{b}.$$

În schimb avem următoarele relații :

$$\overline{a \oplus b} = \bar{a} \oplus b = a \oplus \bar{b} = ab + \bar{a}\bar{b}.$$

Cu alte cuvinte, inversa sumei disjunctive este echivalența

$$\overline{a \oplus b} = (a \equiv b).$$

Pe de altă parte

$$\overline{ab} = a + b = a \oplus b \oplus ab.$$

Alte formule utile :

1) $1 \oplus (1 \oplus a)(1 \oplus b) = a + b$,

2) $a(a \oplus b) = a\bar{b}$,

3) $\bar{a}(a \oplus b) = a(\bar{a} \oplus b) = ab$,

4) $\bar{a}(a \oplus b) = \bar{a}b$,

5) $\bar{a}(\bar{a} \oplus b) = \bar{a}(a \oplus \bar{b}) = \bar{a}\bar{b}$.

Demonstrație. Referitor la formula 1), avem

$$1 \oplus (1 \oplus a)(1 \oplus b) = 1 \oplus \bar{a}\bar{b} = \overline{\bar{a}\bar{b}} = a + b.$$

Faptul că dubla implicație (echivalența) se exprimă prin inversa sumei disjunctive face utilizarea celei din urmă convenabilă în probleme de simplificare ale formulelor care conțin un mare număr de conectori ai dublei implicații.

Exemplu. Fie formula : $f = [\sim((x_1x_2 \equiv x_1x_2) \equiv (x_2y_1 \equiv y_1y_2))] \wedge [\sim(((\sim(x_2x_3 \equiv x_2y_3) \equiv x_2y_3)) \equiv y_2y_3))] \wedge [(x_1x_3 \equiv x_1x_3) \equiv \sim((x_3y_2 \equiv y_1y_3))]$.

Această formulă a fost tratată de către Kapur și Narendran în utilizarea unei metode algebrice de demonstrare automată a teoremelor (1984).

Folosind formulele prezentate, putem scrie succesiv:

$$f = \overline{(x_1(x_2 \oplus y_2)} \oplus \overline{y_1(x_3 \oplus y_3)}\,\overline{(x_3(x_2 \oplus y_2)} \oplus x_2y_3 \oplus y_2y_3)\,\overline{(x_1\overline{(x_3 \oplus y_3)} \oplus y_1(x_3 \oplus y_3))} =$$

$$= (x_2 \oplus y_2)(x_3 \oplus y_3)(\bar{x}_1 \oplus \bar{y}_1)(\bar{x}_1 \oplus \bar{y}_1)(\bar{x}_3 \oplus \bar{y}_3).$$

Dar $(x_3 \oplus y_3)\overline{(x_3 \oplus y_3)} = (x_3 \oplus y_3)\,\overline{(x_3 \oplus y_3)} = 0$.

Deci f este nerealizabilă: $f = 0$.

Principiul rezoluției. Să demonstrăm faptul că următoarea formulă este o tautologie:

$$(1) \qquad (l \oplus A)(\bar{l} \oplus B) \to A \oplus B$$

Avem

$$(l \oplus A)(\bar{l} \oplus B) = (l\bar{A} + \bar{l}A)(\bar{l}\bar{B} + lB) = l\bar{A}B + \bar{l}A\bar{B}$$

$$\overline{l\bar{A}B + \bar{l}A\bar{B}} + \bar{A}B + A\bar{B} = (\bar{l} + A + \bar{B})(l + \bar{A} + B) +$$

$$+ \bar{A}B + A\bar{B} = \bar{l}\bar{A} + \bar{l}B + lA + l\bar{B} + \bar{A}B + A\bar{B} = 1,$$

pentru că dualul

$$(\bar{l} + \bar{A})(\bar{l} + B)(l + A)(l + \bar{B})(\bar{A} + B)(A + \bar{B}) =$$

$$= l(\bar{l} + \bar{A})(\bar{l} + B)(l + \bar{B})(\bar{A} + B)(A + \bar{B}) + A(\bar{l} + \bar{A})(\bar{l} + B)(l + \bar{B}) =$$

$$= (\bar{A} + B)(A + \bar{B}) = l\bar{A}B(\bar{A} + B)(A + \bar{B}) + A\bar{l}(\bar{l} + B)(l + \bar{B})\ B = 0.$$

Consecințele formulei (1) sînt următoarele:

Două clauze scrise în formă de sumă conjunctivă admit un rezolvent de aceeași formă, astfel că nerezolvabilitatea rezolventului antrenează nerezolvabilitatea produsului acestor clauze. Putem reconstitui astfel metoda rezoluției pentru formule FNC (produse de sume disjunctive), în scopul obținerii unei clauze de forma $l \oplus l = 0$, a cărei prezență va indica nerezolvabilitatea formulei inițiale.

Ecuații booleene. Regăsim suma disjunctivă în problema transformării unei ecuații booleene de forma $p = q$ în forma $r = 0$.

Definiție. O pereche de polinoame booleene (p, q) se numește o *ecuație booleană*, iar $(a_1, \ldots, a_n)$ se numește *soluție*, dacă $\bar{p}(a_1, \ldots, a_n) = \bar{q}_B(a_1, \ldots, a_n)$. Un sistem de ecuații este mulțimea de ecuații (p_i, q_i) pentru $1 \leqslant i \leqslant n$, iar soluția sistemului este soluția comună a ecuațiilor (p_i, q_i).

Se obișnuiește scrierea $p = q$ pentru perechea (p, q), cu mențiunea că relația de egalitate întrebuințată nu este una și aceeași cu egalitatea a două polinoame care are loc atunci cînd $\bar{p}_B = \bar{q}_B$ pentru toate evaluările posibile $(a_1, \ldots, a_n)$. A rezolva o ecuație înseamnă a găsi (dacă ele există) evaluările pentru care două polinoame iau aceeași valoare. Ecuația $x + x' = 0$ nu are soluție.

Exemplu. Sistemul de ecuații $p_1 = q_1$, $p_2 = q_2$, unde $p_1 = x_1x_3$, $q_1 = x_2 + x_1x_2x_3'$ și $p_2 = x_1' + x_3$, $q_2 = x_1x_2$, admite soluția $x_1 = x_2 = x_3 = 1$, pentru care $p_1 = q_1$, $p_2 = q_2$.

Teoremă. *Ecuațiile $p = q$ și $pq' + p'q = 0$ au aceleași soluții.*

Fie $(a_1, \ldots, a_n)$ o evaluare a primei ecuații pentru care $\bar{p}(a_1, \ldots, a_n) = a$, iar $\bar{q}(a_1, \ldots, a_n) = b$. Dacă $(a_1, \ldots, a_n)$ este o soluție, $a = b$. Dar

Scanned with OKEN Scanner

$aa' = 0$. Deci şi $(a + b)(a' + b') = 0$ şi $(a + b)(a' + b') = ab' + a'b = 0$, ceea ce arată că $(a_1, \ldots, a_n)$ este o soluţie a ecuaţiei $pq' + p'q = 0$.

Teorema este folosită pentru a transforma un sistem de ecuaţii într-o singură ecuaţie de forma $r = 0$. Fiind dat sistemul (p_i, q_i), $1 \leqslant i \leqslant n$, scriem ecuaţia

$$p_1q_1' + p_1'q_1 + \ldots + p_nq_n' + p_n'q_n = 0.$$

Scriem membrul întîi în forma FNC şi apoi examinăm factorii acestei expresii. Produsul va fi 0, cînd cel puţin unul din factori va fi 0, iar aceasta se produce pentru o evaluare, care constituie soluţie a sistemului dat.

Exemplu. Reluăm exemplul anterior, căutînd soluţiile sistemului

$$x_1x_3 = x_2 + x_1x_2x_3', \qquad x_1' + x_3 = x_1x_2.$$

Scriem o singură ecuaţie de forma sumelor $p_iq_i' + p_i'q_i$:

$$(x_1x_3)(x_2 + x_1x_2x_3')' + (x_1x_3)'(x_2 + x_1x_2x_3') + (x_1' + x_3)(x_1x_2)' + (x_1' + x_3)'(x_1x_2) = 0.$$

Făcînd calculele şi aducînd la forma FNC:

$$(x_1' + x_2' + x_3')(x_1' + x_2 + x_3) = 0.$$

Soluţia este (1, 1, 1) pentru care se anulează factorul $x_1' + x_2' + x_3'$ şi (0, 1, 1) pentru care se anulează factorul $x_1' + x_2 + x_3$.

4.8. DEZVOLTĂRI ALE FUNCŢIILOR BOOLEENE

Reconstituim formele normale ale polinoamelor booleene pornind de la propoziţia:

Fiecare funcţie booleană se poate scrie sub forma

$$f(x) = f(1) \cdot x + f(0) \cdot \bar{x}. \tag{2}$$

Demonstraţie (prin inducţie). Relaţia este adevărată pentru $f(x) = a$, $f(x) = x$, $f(x) = \bar{x}$ pentru că $a = a \cdot x + a \cdot \bar{x}$, $x = 1 \cdot x + 0 \cdot \bar{x}$, $\bar{x} = 0 \cdot x + 1 \cdot \bar{x}$.

Este suficient să arătăm că dacă două funcţii pot fi scrise sub forma (2), atunci suma şi produsul lor este de aceeaşi formă. Într-adevăr,

$$f(x) + g(x) = f(1)x + f(0)\bar{x} + g(1)x + g(0)\bar{x} = (f(1) + g(1))x +$$

$$+ f(0) + g(0))\bar{x}, \quad \text{şi} \quad f(x) \cdot g(x) = (f(1)x + f(0)\,\bar{x})(g(1)x + g(0)\bar{x}) =$$

$$= f(1) \cdot g(1)x + f(0) \cdot g(0)x.$$

Întrucît orice $f(x)$ poate fi construit din elementele lui B cu un număr finit de sume şi produse asupra lui x şi $\bar{x}$, urmează că propoziţia este adevărată. Înseamnă că orice funcţie booleană de o variabilă este de forma $f(x) = ax + b\bar{x}$.

Scanned with OKEN Scanner

A doua propoziție aplică pe prima în cazul a n variabile :

$$f(x_1, \ldots, x_n) = f(1, x_2, \ldots, x_n)\, x_1 + f(0, x_2, \ldots, x_n)\bar{x}_1.$$

Repetînd-o pentru a doua variabilă, se obține

$$f(x_1, \ldots, x_n) = f(1, 1, x_3, \ldots, x_n)\, x_1x_2 + f(1, 0, x_3, \ldots, x_n)\, x_1\bar{x}_2 +$$

$$+ f(0, 1, x_3, \ldots, x_n)\bar{x}_1x_2 + f(1, 1, x_3, \ldots, x_n)\, \bar{x}_1\bar{x}_2.$$

În final, obținem dezvoltarea normală FND pe care o cunoaștem de la citirea tablelor de adevăr și de la reprezentarea normală a polinoamelor booleene :

$$(3)\quad f(x_1, \ldots, x_n = f(1, 1, \ldots 1)x_1x_2 \ldots x_n + f(1, 1, \ldots, 0)x_1x_2 \ldots \bar{x}_n + \ldots$$

$$\ldots + f(0, 0, \ldots, 0)\bar{x}_1\bar{x}_2 \ldots \bar{x}_n.$$

Folosim acum proprietățile sumei disjunctive pentru a scrie compact această dezvoltare.

Definiție. Se numește *complement* al lui x în raport cu a și se notează cu x^a,

$$x^a = xa + \bar{x}\bar{a} = x \oplus \bar{a}.$$

Complementul lui x în raport cu a are proprietatea $x^1 = x$, cînd $a = 1$; $x^a = \bar{x}$, cînd $a = 0$.

Evident $x^{\bar{a}} = x\bar{a} + \bar{x}a = x \oplus a$ și $x^{\bar{1}} = \bar{x}$, cînd $a = 1$; $x^i = x$, cînd $a = 0$.

Avem un mijloc de a scrie unitar un minitermen împreună cu evaluarea funcției care îl precede.

Exemplu. Fie $f(x_1x_2x_3x_4)$. Termenul $f(1, 0, 1, 0)\, x_1x_2x_3x_4$ al dezvoltării normale se poate scrie $f(1, 0, 1, 0)\, x_1^1x_2^0x_3^1x_4^0$ sau $f(a_1a_2a_3a_4)x_1^{a_1}x_2^{a_2}x_3^{a_3}x_4^{a_4}$, unde $a_1 = 1$, $a_2 = 0$, $a_3 = 1$, $a_4 = 0$.

Dezvoltarea (3) devine

$$(4)\qquad f(x_1, \ldots x_n) = \sum_{a_1, \ldots, a_n \in I} f(a_1, \ldots, a_n)\, x_1^{a_1} \ldots x_n^{a_n}$$

unde I este mulțimea celor 2^n evaluări corespunzînd tuturor combinațiilor făcute cu $a_i \in I = \{1, 0\}$.

Reluarea întregului raționament pornind de la reprezentarea unei funcții de o variabilă sub formă de produs

$$f(x) = f(1)x + f(0)\bar{x} = (x + f(0))(\bar{x} + f(1))$$

ne va conduce la forma generală FNC

$$(5)\qquad f(x_1, \ldots, x_n) = \prod_{a_1, \ldots, a_n \in I} (x_1^{\bar{a}_1} + x_2^{\bar{a}_2} + \ldots + x_n^{\bar{a}_n} + f(a_1, \ldots, a_n)).$$

Scanned with OKEN Scanner

Exemplu. Fie o funcție definită astfel : $f(a_1a_2a_3) = 0$ pentru (1, 0, 1), (1, 0, 0) și (0,0,1) și $f(a_1a_2a_3) = 1$ pentru celelalte evaluări. Atunci formula (4) oferă următoarea dezvoltare a lui f :

$$f(x_1, x_2, x_3) = f(1, 1, 1)\, x_1^1x_2^1x_3^1 + f(1, 1, 0)\, x_1^1x_2^1x_3^0 + f(0, 1, 1)\, x_0^1x_2^1x_3^1 + f(0,1,0)x_1^0x_2^1x_3^0 +$$
$$+ f(0, 0, 0)x_1^0x_2^0x_3^0 = x_1x_2x_3 + x_1x_2\bar{x}_3 + \bar{x}_1x_2x_3 + \bar{x}_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2\bar{x}_3.$$

Pentru obținerea produsului, recurgem la formula (5) :

$$f(x_1, x_2, x_3) = (\bar{x}_1 + x_2 + \bar{x}_3 + f(1, 0, 1))\,(\bar{x}_1 + x_2 + x_3 + f(1, 0, 0))\,(x_1 + x_2 + \bar{x}_3 +$$
$$+ f(0, 0, 1)) = (\bar{x}_1 + x_2 + \bar{x}_3)\,(\bar{x}_1 + x_2 + x_3)(x_1 + x_2 + \bar{x}_3).$$

Factorii în care $f(a_1, a_2, a_3) = 1$ nu mai apar întrucît devin 1.

Dezvoltările (4) și (5) se numesc *dezvoltări Lagrange*, iar în literatură miniterменii sînt numiți uneori *monomi Lagrange*. Dezvoltările Lagrange sînt normale, fiecărei funcții corespunzîndu-i o singură dezvoltare de această formă.

În expresia normală FND a unei funcții booleene putem înlocui pe $+$ cu $\oplus$ fără ca formula să-și piardă valabilitatea. Faptul se datorează trecerii de la disjuncție la sumă disjunctivă prin formula $x+y = x\oplus y\oplus \oplus xy$. Dar orice produs de miniterмeni din dezvoltarea normală au produsul 0 și atunci cînd aplicăm formula de transformare, nu rămîn decît sumele disjunctive.

Exemplu. $f(x) = f(1)x + f(0)\bar{x} = f(1)x + f(0)\bar{x} + f(1)f(0)x\bar{x} = f(1)x + f(0)\bar{x}$.

Convenim să scriem simbolul $\sum_{\oplus}$ pentru o sumă de termeni legați de suma conjunctivă.

Dezvoltarea lui Lagrange corespunzătoare pentru orice $f(x_1, \ldots, x_n)$ va fi $f(x_1, \ldots, x_n) = \sum\limits_{\substack{\oplus \\ a_1,\ldots,a_n \in I}} f(a_1, \ldots, a_n)x_1^{a_1} \ldots x_n^{a_n}$.

Dezvoltări cvasinormale. În cazul în care orice funcție $f(x_1, \ldots, x_n)$ poate fi scrisă ca o funcție echivalentă de o formă dată, fără ca această reprezentare să fie unic determinată, vom numi-o pe aceasta din urmă *formă cvasinormală*.

Să stabilim formula :

(6) $$f(x, y) = f(x, x)(x \oplus \bar{y}) + f(x, \bar{x})\,(x \oplus y) \quad \text{sau}$$

$$f(x, y) = f(x, x)(xy + \bar{x}\bar{y}) + f(x, \bar{x})\,(x\bar{y} + \bar{x}y),$$

valabilă pentru orice funcție booleană $f(x, y)$.

Demonstrație. Pornim de la

$$f(x, x) = f(1, 1)\,x + f(0, 0)\,\bar{x}, \qquad f(x, \bar{x}) = f(1, 0)\,x + f(0, 1)\bar{x}.$$

Înmulțind cele două relații pe rînd cu x și $\bar{x}$, obținem

$$f(x, x)\,x = f(1, 1)x, \qquad f(x, x)\bar{x} = f(0, 0)\bar{x},$$

$$f(x, \bar{x})x = f(1, 0)x, \qquad f(x, \bar{x})\bar{x} = f(0, 1)\bar{x}.$$

Folosim dezvoltarea

$$f(x, y) = f(1, 1)xy + f(1, 0)x\bar{y} + f(0, 1)\bar{x}y + f(0, 0)\bar{x}\bar{y}$$

Scanned with OKEN Scanner

și, înlocuind cu (6),

$$f(x, y) = f(x, x)xy + f(x, \bar{x})\, x\bar{y} + f(x, \bar{x})\, \bar{x}y + f(x, x)\, \bar{x}\bar{y}.$$

Grupînd termenii 1 cu 4, 2 cu 3, se obține

$$f(x, y) = f(x, x)\,(xy + \bar{x}\bar{y}) + f(x, \bar{x})\,(x\bar{y} + \bar{x}y).$$

Formulele se pot generaliza. Dăm forma cvasinormală a unor funcții de 3 și 4 variabile :

$$f(x, y, z) = f(x, x, x)(x \oplus \bar{y})(x \oplus \bar{z}) + f(x, \bar{x}, x)\,(x \oplus y)(x \oplus \bar{z}) +$$

$$+ f(x, x, \bar{x})(x \oplus \bar{y})\,(x \oplus z) + f(x, \bar{x}, \bar{x})\,(x \oplus y)\,(x \oplus z),$$

$$f(x, y, z, u) = f(x, x, x, x)(x \oplus \bar{y})(x \oplus \bar{z})(x \oplus \bar{u}) +$$

$$+ f(x, \bar{x}, x, x)\,(x \oplus y)(x \oplus \bar{z})\,(x \oplus \bar{u}) +$$

$$+ f(x, x, \bar{x}, x)\,(x \oplus \bar{y})\,(x \oplus z)\,(x \oplus \bar{u}) +$$

$$+ f(x, x, x, \bar{x})(x \oplus \bar{y})\,(x \oplus \bar{z})\,(x \oplus u) +$$

$$+ f(x, x, \bar{x}, \bar{x})\,(x \oplus \bar{y})\,(x \oplus z)\,(x \oplus u) +$$

$$+ f(x, \bar{x}, x, \bar{x})\,(x \oplus y)\,(x \oplus \bar{z})\,(x \oplus u) +$$

$$+ f(x, \bar{x}, \bar{x}, x)\,(x \oplus y)(x \oplus z)\,(x \oplus \bar{u}) +$$

$$+ f(x, \bar{x}, \bar{x}, x)(x \oplus y)(x \oplus z)(x \oplus \bar{u}).$$

Formulele cvasinormale de mai sus au fost obținute, considerînd variabila x privilegiată. Alte dezvoltări ale aceleiași funcții se obțin producînd dezvoltări analoage în raport cu oricare altă variabilă.

Forma cvasinormală considerată are avantajul de a conține 2^{n-1} termeni, jumătate din numărul termenilor dezvoltării Lagrange. Acest lucru sugerează folosirea ei pentru un algoritm de testare a rezolvabilității unei formule. Numim metoda ce rezultă *metoda diagonalei*.

Aplicație. 1) Cazul $n = 2$. Din formulele (6) rezultă că f este tautologie dacă și numai dacă

$$f(x, x) = 1, \qquad f(x, \bar{x}) = 1.$$

Într-adevăr, f este tautologie cînd

$$f(1, 1) = f(1, 0) = f(0, 1) = f(0, 0) = 1,$$

ceea ce, conform relațiilor de mai înainte, este posibil dacă

$$f(x, x) = 1, \qquad f(x, \bar{x}) = 1.$$

Scanned with OKEN Scanner

La fel, f este nerealizabilă cînd

$$f(1, 1) = f(1, 0) = f(0, 1) = f(0, 0) = 0,$$

ceea ce este echivalent, datorită acelorași relații, cu

$$f(x, x) = f(x, \bar{x}) = 0.$$

În cazul în care $f(x, x) \neq f(x, \bar{x})$ funcția este realizabilă. Valorile pe care $f(x, x)$ și $f(x, \bar{x})$ pot să le ia sînt 0, 1, x, $\bar{x}$.

În locul tablei de adevăr care reclamă patru încercări avem nevoie în metoda diagonalei doar de două.

Exemple. a) $f = (p + q)(p + \bar{q})(\bar{p} + q)(\bar{p} + \bar{q})$ este nerealizabilă pentru că $f(p, p) = f(p, \bar{p}) = p\bar{p} = 0$.

b) $f = \sim q \wedge (p \to q) \to \sim p$ este o tautologie pentru că $f(p, q) = \bar{q}(\bar{p} \vee q) \to \bar{p}$, iar $f(p, p) = \bar{p} \to \bar{p} = 1$ și $f(p, \bar{p}) = 0 \to \bar{p} = 1$.

Cazul $n = 3$. Considerăm acum formula $f(x, y, z)$ căreia îi aplicăm (6):

$$f(x, y, z) = f(x, x, z)(x \oplus \bar{y}) + f(x, \bar{x}, z)(x \oplus y).$$

Formula $f(x, y, z)$ este tautologie (contradicție) dacă și numai dacă $f(x, x, z)$ și $f(x, \bar{x}, z)$ sînt tautologii (contradicții). Repetăm aplicarea formulei (6) la fiecare dintre ele și obținem

$$f(x, x, z) = f(x, x, x)(x \oplus \bar{z}) + f(x, x, \bar{x})(x \oplus z),$$

$$f(x, x, x) = f(x, x, z)(x \oplus \bar{z}) + f(x, \bar{x}, \bar{x})(x \oplus z);$$

$f(x, y, z)$ este tautologie, dacă

$$f(x, x, x) = f(x, x, \bar{x}) = f(x, \bar{x}, x) = f(x, \bar{x}, \bar{x}) = 1$$

și este contradicție (nerealizabilă), dacă

$$f(x, x, x) = f(x, x, \bar{x}) = f(x, \bar{x}, x) = f(x, \bar{x}, \bar{x}) = 0.$$

Exemplu. a) Formula $f(x, y, z) = (\bar{x} + \bar{y} + \bar{z})(x + y + z)$ este realizabilă. Se observă $f(x, x, x) = 0$, dar $f(x, \bar{x}, x) = 1$. Nu este nevoie să mai continuăm.

b) Fie relația $f(l, A, B) = (l \oplus A)(\bar{l} \oplus B) \to A \oplus B$.

Prin aceeași metodă se demonstrează că relația este tautologie.

Cazul $n = 4$. Tabla de valori cere $2^4 = 16$ încercări, care pot fi înlocuite cu 8. Combinațiile variabilelor pentru care căutăm valorile

Scanned with OKEN Scanner

lui $f(x, y, z, u)$ sînt date de tablourile

	x	x	x
x	x	x	x
x	x	$\bar{x}$	x
x	x	x	$\bar{x}$

	x	x	x
x	x	$\bar{x}$	$\bar{x}$
x	$\bar{x}$	x	$\bar{x}$
x	$\bar{x}$	$\bar{x}$	x

de unde vine denumirea metodei de „diagonală".

Exemple. a) Fie

$$P(x, y, z, u) = (x + y + z + u)(\bar{x} + \bar{y} + \bar{z} + \bar{u})(\bar{x} + \bar{y} + z + u).$$

Avem $P(x, x, x, x) = 0$, $P(x, x, x, \bar{x}) = 1$, deci P este realizabilă.

b) Fie $P(x, y, z, u) = (x + y + z + u)(\bar{x} + \bar{y} + \bar{z} + \bar{u})(\bar{x} + \bar{y} + \bar{z} + u)(x + y + z + \bar{u})(x + y + \bar{z} + u)(\bar{x} + \bar{y} + z + \bar{u})(x + \bar{y} + z + u)(\bar{x} + y + \bar{z} + \bar{u})(\bar{x} + y + z + u)(x + \bar{y} + \bar{z} + u)$.

Deși $P(x, x, x, x) = P(\bar{x}, x, x, \bar{x}) = P(x, x, \bar{x}, x) = P(x, \bar{x}, x, x) = P(x, \bar{x}, \bar{x}, \bar{x}) = 0$, a șasea substituție conduce la $P(x, x, \bar{x}, \bar{x}) = 1$ și deci P este realizabilă.

Reprezentări de funcții prin operatori ternari. Numeroase probleme și calcule logice sînt facilitate prin introducerea unor operatori ternari, spre deosebire de cei binari folosiți în exclusivitate pînă la acest punct.

Se numește *operația majoritate* (sau *funcția majoritate*) următoarea aplicație $f : B^3 \to B$:

$$f(x, y, z) = xy + yz + zx.$$

În scriere prefix ea este $Mxyz$ și corespunde conectorului logic ternar neutilizat pînă acum:

$$(x \wedge y) \vee (y \wedge z) \vee (x \wedge z) = (x \vee y) \wedge (y \vee z) \wedge (x \vee z).$$

Operația majoritate are deci proprietatea

$$Mxyz = xy + yz + zx = (x + y)(y + z)(z + x).$$

De asemenea, avem

$$\begin{aligned} x &= 0, & M0yz &= yz \\ x &= 1, & M1yz &= y + z \\ y &= x, & Mxxz &= x \\ y &= x', & Mxx'z &= z \\ x &= y = z, & Mxxx &= x \end{aligned}$$

Cu ajutorul funcției majoritate, funcțiile de o singură variabilă se pot reprezenta astfel:

1) $f(x) = MMxf(0)\,1\,Mx'f(1)10$ sau

1') $f(x) = MMxf(1)\,0\,Mx'f(0)0\,1 = MMx'f(0)0\,Mx'f(1)\,1x =$

$= MM\,xf\,(1)0\;Mxf\,(0)1\;x'.$

Scanned with OKEN Scanner

Într-adevăr pentru formula 1) :

$$x = 0,\ Mx'f(1)1 = 1,\ M\,0\,f(0)\,1 = f(0) \text{ și } M\,f(0)1\,0 = f(0),$$

$$x = 1,\ M\,f(0)\,1 = 1,\ M\,0\,f(1)\,1 = f(1),\ M\,1f(1)0 = f(1).$$

Pentru două variabile :

$$f(x, y) = MM\,xy\,f\,(x, x')\,Mx'y\,f(x, x)\,y' =$$

$$= MM\,x'y'\,f(x, x')\,Mxy'\,f(x, x)y =$$

$$= MMy'x'f(x, x')\,Myz'f(x, x)\,x =$$

$$= MM\,yx\,f(x, x')\,M\,y'x\,f(x, x)\,x'.$$

Din nou, pentru

$$y = x,\quad f(x, x) = Mx\,f(x, x)\,x' = f(x, x),$$

$$y = x',\quad f(x, x') = M\,f(x, x')x'x = f(x,x').$$

Pentru trei variabile este valabilă formula [7]

$$f(x, y, z) = MMf(x, x, z)\,y'x\,Mf(x, y, y)\,z'y\,M\,f(z, y, z)x'z.$$

În conformitate cu metoda diagonalei o verificăm pentru

$$f(x, x, x) = Mf(x, x, x)\,f(x, x, x)\,f(x, x, x) = f(x, x, x),$$

$$f(x, x, x') = M\,f(x, x, x')\,xx' = f(x, x, x')$$

$$f(x, x', x) = Mxx'\,f(x, x', x') = f\,(x, x', x),$$

$$f(x, x', x') = M\,x\,f(x, x', x')x' = f(x, x', x').$$

Interesul acestei reprezentări constă în exprimarea lui $f\,(x, y, z)$ în funcție de trei funcții : $f(x, x, z)$, $f(x, y, y)$ și $f(z, y, z)$ în loc de opt instanțieri presupuse de către dezvoltarea normală, sau patru cîte intervin în metoda diagonalei.

Aplicație. a) Să se demonstreze că $f(x, y, z) = (x \to y)\,(y \to z) \to (x \to z)$ este o tautologie :

$$f(x, x, z) = 1,\quad f(x, y, y) = 1,\quad f(z, y, z) = 1$$

$$f(x, y, z) = M'(y' + x)\,(z' + y)\,(x' + z) = 1$$

b) Formula $f(p, q, r) = (p + q)\,(p + q' + r)\,(p + q' + r)\,(p' + r)(p' + r')$ este nerealizabilă :

$$f(p, p, r) = p\,1\,1'\,(p' + r)(p' + r') = 0,$$

$$f(p, q, q) = (p + q)\,1(p + q')(p' + q)(p' + q') = 0,$$

$$f\,(r, q, r) = (r + q)(r + q')\,1\,(r + q') = 0,$$

$$MM\,O\,q'p\,M\,O\,r'qMOp'r = M(q'p)\,(r'q)\,(p'r) = 0.$$

Operatorul U și algebra ITE. Un alt operator ternar care și-a găsit importante aplicații este

$$Uxyz = xy + x'z = (x + z)(x' + y) = (x \to y)(x + z).$$

Se observă că:

$$x = 1 \qquad U\,1\,yz = y$$

$$x = 0 \qquad U\,0\,yz = z.$$

O funcție de o variabilă poate fi reprezentată cu ajutorul lui U:

$$f(x) = Ux\,f(1)\,f(0) = f(1)x + x'\,f(0),$$

iar o funcție de n variabile:

$$f(X_n) = Ux_n f(X_{n-1}, 1)\,f(X_{n-1}, 0),$$

unde am notat cu X_n variabilele $x_1, x_2, \ldots, x_n$,

Pentru $n = 3$ avem formula

$$f(x, y, z) = Ux\,Uy\,Uz\,f(1, 1, 1)\,f(1, 1, 0)\,Uz\,f(1\,0, 1)\,f(1, 0, 0)\,Uy\,Uz\,f(0, 1, 1)\cdot$$
$$\cdot f(0, 1, 0)\,Uz\,f(0, 0, 1)\,f(0, 0, 0)$$

$Uxyz$ fiind egal cu $(x \to y)\,(x + z)$, se poate citi „dacă x, atunci y, altfel z". De aceea algebra construită cu această operație se numește *algebră ITE* (If x, Then y, Else z).

Algebra $\langle A, U, 0, 1\rangle$, unde $U : A^3 \to A$, se numește *algebră ITE* dacă pentru orice $x, y, z, u, v \in A$ sînt satisfăcute condițiile:

1) $U(U(x, y, z), u, v) = U(x, U(y, u, v), U(z, u, v))$,

2) $U(x, U(y, z, u), v) = U(y, U(x, z, v), U(x, u, v))$,

3) $U(x, z, U(y, u, v)) = U(y, U(x, z, u), U(x, z, v))$,

4) $U(x, U(x, y, z), u) = U(x, y, u)$,

5) $U(x, y, U(x, z, u)) = U(x, y, u)$,

6) $U(x, 1, 0) = x$,

7) $U(x, 0, 0) = 0$, $U(x, 1, 1) = 1$,

8) $U(0, x, y) = y$, $U(1, x, y) = x$.

Orice algebră Boole este o algebră ITE. Este suficient să luăm $U(x, y, z) = (x \cap y) \cup (x' \cap z)$. Reciproc, dacă $\langle A, U, 0, 1\rangle$ este o algebră ITE, punînd $x \cup y = U(x, 1, y)$, $x \cap y = U(x, y, 0)$ și $x' = U(x, 0, 1)$ și $x' = U(x, 0, 1)$, se obține o algebră booleană.

Dezvoltări Reed-Müller. Operația $+$ poate fi folosită în dezvoltările funcțiilor booleene. Fie funcția de o singură variabilă

$$f(x_1) = d_0 x_1 + d_1 x_1$$

(pentru o scriere convenabilă, folosim $\bar{x}$ pentru x') sau

$$f(x_1) = \overline{d_0 \bar{x}_1 \cdot d_1 x_1}.$$

Scanned with OKEN Scanner

Introducem $\bar{x} = x \oplus 1$ și utilizăm proprietățile stabilite la 4.7. Atunci

$$f(x_1) = (d_0(x_1 \oplus 1) \oplus 1)(d_1 x_1 \oplus 1) \oplus 1 = d_0 d_1 x_1 \oplus d_0 d_1 x_1 \oplus$$

$$\oplus d_0 x_1 \oplus d_0 \oplus d_1 x_1 = d_0 \oplus (d_0 \oplus d_1) x_1 = c_0 \oplus c_1 x_1.$$

O asemenea dezvoltare se numește *dezvoltare Reed-Müller*. Coeficienții $c_0 = d_0$, $c_1 = d_0 \oplus d_1$ sînt 0 sau 1 și pot fi legați de coeficienții $d_i (i = 1, 2)$ prin matricea (modulo 2)

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \end{bmatrix} \quad \text{sau} \quad C = T_1 D$$

Cazul $n = 2$ se tratează analog:

$$f(x_2, x_1) = d_0 m_0 + d_1 m_1 + d_2 m_2 + d_3 m_3 =$$

$$= d_0 m_0 \oplus d_1 m_1 \oplus d_2 m_2 \oplus d_3 m_3 = d_0 \bar{x}_2 \bar{x}_1 \oplus d_1 \bar{x}_2 x_1 \oplus d_2 x_2 \bar{x}_1 \oplus d_3 x_2 x_1 =$$

$$= d_0 (x_2 \oplus 1)(x_1 \oplus 1) \oplus d_2 (x_2 \oplus 1) x_1 \oplus d_2 x_2 (x_1 \oplus 1) \oplus d_3 x_2 x_1 =$$

$$= d_0 \oplus (d_0 \oplus d_1) x_1 \oplus (d_0 \oplus d_2)\, x_2 \oplus (d_0 \oplus d_1 \oplus d_2 \oplus d_3)\, x_2 x_1 =$$

$$= c_0 \oplus c_1 x_1 \oplus c_2 x_2 \oplus c_3 x_2 x_1,$$

unde

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad \text{sau} \quad C = T_2 D.$$

Se observă că

$$T_2 = \begin{bmatrix} T_1 & 0 \\ T_1 & T_1 \end{bmatrix}$$

În cazul general, dezvoltarea Reed-Müller a unei funcții de n variabile este

$$f(x_n, x_{n-1}, \ldots, x_1) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_2 x_1 + c_4 x_3 +$$

$$+ c_5 x_3 x_1 + \ldots + c_{2^n - 1} x_n x_{n-1} \ldots x_1,$$

ceea ce se mai scrie

$$f(x_n, x_{n-1}, \ldots, x_1) = \sum_{i=0}^{2^n - 1}{}_{\oplus}\, c_i\, x_n^{e_{i,n}} x_{n-1}^{e_{i,n-1}} \ldots x_1^{e_{i,1}}$$

Semnul $\sum_{\oplus}$ reprezintă o sumă $\sum$ de termeni legați prin $\oplus$, e_{ij} sînt 0 sau 1, iar $x_k^0 = 1$, $x_k^1 = x_k$. Dezvoltarea are 2^n termeni și se mai notează

$$f(x_n, x_{n-1}, \ldots, x_1) = \sum_{i=0}^{2^n - 1} c_i \pi_i.$$

Scanned with OKEN Scanner

Corespondenții minitermenilor m_i din dezvoltarea normală sînt într-o dezvoltare Reed-Müller, termenii $\pi : \pi_i$ Indicele i este echivalentul zecimal al numărului binar format din $e_{i,n}, \ldots e_{i,1}$. De exemplu :

$$x_6 x_3 x_1 = x_6^1 x_5^0 x_4^0 x_3^0 x_2^0 x_1^1 = \pi_{37}$$

și

$$\pi_{23} = x_5^1 x_4^0 x_3^1 x_2^1 x_1^1 = x_5 x_3 x_2 x_1.$$

4.9. COMPLEXITATE

În acest capitol au fost prezentați o serie de algoritmi „clasici" pentru rezolvarea unor probleme din calculul propozițional. Algoritmii lui Wang, Davis-Putnam și rezoluția rezolvă în principiu aceeași problemă : a decide dacă o formulă este realizabilă (satisfiabilă) sau nu.

Putînd recurge la mai mulți algoritmi pentru soluționarea aceleiași probleme, care sînt criteriile pentru alegerea „celui mai bun" dintre ei, pentru a fi transpus pe calculator? Există multe posibilități de măsurare a performanțelor unui algoritm, după timpul de execuție, memoria utilizată, lungimea algoritmului transpus într-un limbaj de programare etc.

Analiza logaritmilor nu intră în preocupările acestui volum. Ne vom rezuma la prezentarea cîtorva idei generale legate strict de algoritmii incluși în acest capitol.

Problema satisfiabilității (realizabilității) unei formule FNC, pe scurt SAT, este problema de a determina dacă o formulă în CP la FNC admite o realizare pentru o anumită combinație de valori 0, 1 atribuite variabilelor propoziționale. Pentru o formulă cu n variabile timpul de execuție a oricărui algoritm cunoscut care rezolvă SAT este exponențial (proporțional cu 2^n). Chiar pentru valori relativ mici ale lui n timpul de execuție exponențial implică imposibilitatea practică de a decide satisfiabilitatea oricărei formule de n variabile.

O problemă este desemnată ca fiind „tratabilă" dacă este cunoscut un algoritm polinomial pentru aceasta. Se obișnuiește să se noteze cu **P** clasa acestor probleme polinomiale.

O problemă de importanță deosebită în informatica teoretică actuală este de a găsi algoritmi eficienți cu timpi de execuție proporționali cu polinoame în n de grad cît mai mic.

În această excepție, problema SAT este intratabilă. Una din cele mai celebre probleme deschise, pe care le are de rezolvat matematica informaticii, problemă cunoscută sub numele de „**P = NP?**", este de a infirma sau confirma apartenența sigură a problemei SAT la clasa problemelor tratabile.

Ceea ce justifică interesul deosebit față de această problemă nu este numai importanța intrinsecă pentru demonstrarea automată și pentru inteligența artificială a eventualei descoperiri a unui algoritm „bun" (polinomial) pentru SAT, ci faptul că o dată cu el, o gamă foarte largă de probleme „grele" din domenii variate de aplicabilitate, ar deveni simultan „tratabile". Cîteva exemple : probleme de grafuri (problema voiajorului comercial, a drumului hamiltonian etc.), de proiectare optimă de trasee etc.

Scanned with OKEN Scanner

Mai mult chiar, orice problemă care aparţine unei clase speciale **NP** (de la nedeterminist — polinomial) are proprietatea că ar deveni tratabilă o dată cu SAT.

Ulterior descoperirii acestui fapt (în 1971, de către Stephen Cook) s-a demonstrat că şi foarte multe alte probleme sînt „echivalente" cu SAT, în sensul că un algoritm „bun" pentru ele ar implica un algoritm „bun" pentru toate problemele din **NP** (direcţie iniţiată în 1972 de Richard Karp). Aceste probleme, care aparţin lui **NP**, dar sînt echivalente cu SAT în sensul de mai sus, se numesc **NP**-*complete*.

Faptul că, în ciuda eforturilor multor specialişti în proiectarea algoritmilor şi în ciuda gamei largi de probleme echivalente abordate, nu s-a descoperit pînă la ora actuală nici un algoritm polinomial pentru o problemă **NP**-completă, pare să încline în favoarea opiniei că $\mathbf{P} \neq \mathbf{NP}$, adică un astfel de algoritm nu există ! Dar pînă la ora actuală „$\mathbf{P} = \mathbf{NP}$?" rămîne o problemă deschisă.

În ciuda pesimismului deductibil dintr-o concluzie de tip $\mathbf{P} \neq \mathbf{NP}$, demonstrarea automată are şanse de utilizare practică datorită următorului fapt : pentru clase restrînse de formule se pot pune în evidenţă algoritmi polinomiali de rezolvare a satisfiabilităţii.

Cîteva exemple :

2SAT : problema de a decide satisfiabilitatea pentru clauze cu cel mult 2 literali pe clauză — este polinomial decidabilă.

HORN : cînd clauzele conţin cel mult un literal nenegat, ele poartă denumirea de clauze HORN. În acest caz, SAT este decidabilă în timp polinomial (chiar liniar — deci proporţional cu n).

Cu titlu informativ să amintim că 3SAT, problema SAT pentru clauze cu cel mult 3 literali pe clauză, este **NP**-completă, deci tot atît de grea ca şi SAT.

4.10. NOTĂ ISTORICĂ

Primul eveniment al demonstrării automate a teoremelor s-a petrecut în 1957, cînd A. Newell, J. C. Show şi H. A. Simon au comunicat că programul lor LT pentru maşina Johnniac a demonstrat 38 din primele 52 teoreme din *Principia Mathematica* de Bertrand Russell şi Alfred Whitehead. În 1958, Wang Hao a demonstrat toate teoremele din primele cinci capitole (peste 200) în 37 minute, din care cele 52 teoreme abordate anterior în 5 minute. În exemplele date se reproduce demonstraţia teoremelor 2.45, 5.21 şi 4.45 din *Principia Mathematica*. Extinzînd metoda sa, Wang a reuşit să demonstreze şi cele 150 de teoreme ale calculului cu predicate din următoarele cinci capitole din opera lui Russell şi Whitehead. Meritul lui Wang a fost răsplătit în 1984 de Premiul pentru demonstrarea automată a teoremelor.

Ca şi procedeul lui Wang, regulile Davis-Putnam urmăresc reducerea unei formule la forme mai simple (1960). Prin Wang şi Davis-Putnam ilustrăm procedeele nerezolutive. În 1965 J. A. Robinson introduce metoda rezoluţiei care domină, timp de aproape două decenii, aproape toate metodele de demonstrare automată, fiind variante şi rafinări ale principiului rezoluţiei.

Scanned with OKEN Scanner

Metodele ilustrate aici au fost folosite într-o formă extinsă pentru calculul cu predicate, întrucît teoremele acestui calcul sînt adevărate pietre de încercare ale demonstrării automate. După o perioadă de stagnare, provenită din dificultatea de a stăpîni fenomenul proliferării clauzelor, în anul 1980 creşterea performanţelor calculatoarelor şi recursul la metodele algebrice au dat un nou impuls demonstrării automate a teoremelor. Dacă, pînă în deceniul nostru, metodele serveau în special la demonstrarea automată a unor teoreme cunoscute (ca cele din teoria grupurilor), în ultimii ani au fost obţinute teoreme noi şi au fost rezolvate probleme deschise în matematică.

4.11. EXERCIŢII ŞI PROBLEME

1. Să se cerceteze prin algoritmul lui Wang validitatea formulelor din CP :

a) $(p \to q) \to ((p \wedge r) \to (q \wedge r))$,

b) $(p \equiv q) \to ((p \to r) \to (q \to r))$,

c) $((p \to q) \to r) \to ((r \to p) \to (s \to p))$.

2. Să se aplice procedeul Davis-Putnam pentru a stabili corectitudinea deducţiei

$$p \to q,\ r \vee \bar{q},\ \overline{p \wedge r} \models \bar{p}.$$

3. Să se folosească regulile Davis-Putnam pentru a stabili dacă următoarele formule sînt sau nu realizabile :

a) $(p \vee q) \wedge (r \vee q) \wedge r \wedge q$,

b) $(p \vee q) \wedge (p \vee q) \wedge r$,

c) $(p \vee q) \wedge \bar{q} \wedge (\bar{p} \vee q \vee \bar{r})$.

4. Să se aplice principiul rezoluţiei la studiul mulţimilor de clauze :

a) $S = \{\bar{p} \vee \bar{q} \vee r,\ p \vee r,\ q \vee r,\ \bar{r}\}$

b) $S = \{p \vee q, q \vee r, r \vee w, \bar{r} \vee \bar{p}, \bar{w} \vee \bar{q}, \bar{q} \vee \bar{r}\}$,

c) $S = \{p \vee q \vee r, \bar{p}, \bar{q}, \bar{r} \vee w, \bar{w}\}$,

d) $S = \{p \vee q, p \vee \bar{q} \vee r, p \vee \bar{q} \vee \bar{r}, \bar{p} \vee r, \bar{p} \vee \bar{r}\}$.

5. Să se întocmească diagramele lui Karnaugh pentru cele 16 funcţii polinomiale în B cînd $n = 2$, pornind de la tabloul conectorilor binari (v. 1.11) şi să se utilizeze pentru simplificarea lor.

6. Să se simplifice

$$f = xyzuv + xyz'uv + xy'zu'v + x'yz'uv + xy'zu'v + x'yz'u'v' + x'yzuv.$$

7. Să se determine implicaţii primi ai funcţiei

$$f(x, y, z, u) = xyz + xyz' + x'y'u + yzu.$$

8. Să se stabilească forma minimală a funcţiilor

$$f = z(y + u) + yu' + y'z'u,$$

$$g = x'y' + xzu + xyu' + y'z,$$

folosindu-se algoritmul Quine-McCluskey.

9. Care este forma minimală a funcției

$$f = wx'y'z + w'xy'z' + wx'y'z' + w'xyz + w'x'y'z' + wxyz + wx'yz + w'xyz' + w'x'yz'?$$

10. Să se stabilească caracterul de tautologii ale formulelor T1—T12, recurgînd la algoritmi algebrici.

11. Să se stabilească algebric echivalențele din CP:

a) $A \vee AB \Leftrightarrow A$,

b) $A(A \vee B) \Leftrightarrow A$,

c) $A \vee \bar{A}B \Leftrightarrow A \vee B$,

d) $A(\bar{A} \vee B) \Leftrightarrow AB$,

e) $(A \vee B)(\bar{A} \vee C)(B \vee C) \Leftrightarrow (A \vee B)(\bar{A} \vee C)$,

f) $AB \vee \bar{A}C \vee BC \Leftrightarrow AB \vee \bar{A}C$

12. Să se demonstreze că într-o algebră ITE, oricare ar fi aplicația $f: A^n \to A$, pentru orice $x, y_i, z_i \in A$, $i = 1, \ldots, n$, are loc relația

$$f(U(x, y_1, z_1), \ldots, U(x, y_m, z_n)) = U(x, f(y_1, \ldots, y_n), f(z_1, \ldots, z_n)).$$

13. Să se simplifice prin diagramele lui Karnaugh:

a) $xyz' + x'yz' + (x + y' + z')' + (x + y + z)' + z(x' + y)$,

b) $(x' + y)(z + u')(y + u'x)$.

14. Ce diagrame Karnaugh au următoarele polinoame:

a) $(x + y)(y + z)(z + u)$,

b) $xyu + (x' + y)z$?

BIBLIOGRAFIE

1. Calude, C. *Complexitatea calculului.* București, Editura Științifică și Enciclopedică, 1982.
2. Carvallo, M. *Principes et application de l'analyse booléene.* Paris, Gauthier-Villars, 1965.
3. Chang, C. L., Lee, R. C. T. *Symbolic Logic and Mechanical Theorem Proving.* New York, Academic Press, 1973.
4. Davis, M., Weyuker, E. *Computability, Complexity and Languages.* New York, Academic Press, 1983.
5. Denis-Papin, M., Malgrange, Y. *Exerciții de calcul boolean cu soluțiile lor.* București, Editura Tehnică, 1970.
6. Hohn, F. *Applied Boolean Algebra.* New York, MacMillan, 1966.
7. Kuntzmann. J. *Algèbre de Boole.* Paris, Dunod. 1965.
8. Lidl, R., Pilz, G. *Applied Abstract Algebra.* New York, Springer Verlag, 1984.
9. Stănășilă, O. *Noțiuni și tehnici de matematică discretă.* București, Editura Științifică și Enciclopedică, 1985.
10. Wang, H. *Studii de logică matematică.* București, Editura Științifică, 1972.

Scanned with OKEN Scanner

5.
CIRCUITE LOGICE

5.1. NOȚIUNI INTRODUCTIVE

O clasă largă de aplicații ale algebrei lui Boole este dată de *calculul circuitelor cu contacte*. Să presupunem că avem un circuit electric cu comutatoare, în care prezența sau absența curentului este determinată cu pozițiile comutatoarelor : închis sau deschis. Acest fapt a dat sugestia unei interpretări a algebrei booleene, în care elementele algebrei sînt contactele cu stările 1 și 0, iar cele două operații $\wedge$, $\vee$ sînt legăturile în paralel și în serie. Un *contact* este reprezentat pe diagrama circuitului ca un bloc (fig. 5.1, a). Curentul trece cînd $v(x) = 1$ și nu trece cînd $v(x) = 0$. Faptul că un contact poate avea aceste două stări, *închis* și *deschis*, explică numele de *bipozițional*.

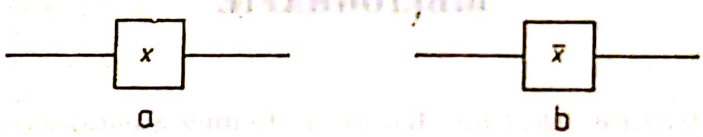


Fig. 5.1

Simbolul din fig. 5.1, b reprezintă un contact care este deschis cînd x este închis și este închis cînd x este deschis.

Figura 5.2 reprezintă o legătură în serie, întrucît curentul trece numai cînd ambele contacte sînt închise, cele două blocuri în serie vor reprezenta produsul $x \wedge y$. Legătura paralelă din fig. 5.3 reprezintă suma $x + y$, întrucît curentul trece cînd cel puțin unul din contacte este închis.

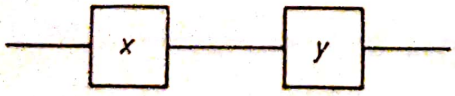


Fig. 5.2

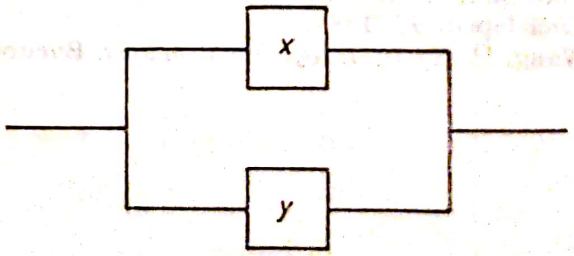


Fig. 5.3

Un circuit este o combinație de legături în serie și paralel care leagă un punct de intrare de un punct de ieșire. Unui circuit îi corespunde o *funcție de transmisie* care ia valoarea 1 cînd curentul trece de la punctul de intrare la cel de ieșire și 0 cînd curentul nu trece.

Scanned with OKEN Scanner

Circuitul din fig. 5.4 va avea ca funcție de transmisie $T = xy' + (x' + y)z$. Dar această funcție se poate reduce la o formă mai simplă, $T = xy' + z$, căreia îi corespunde un circuit mai simplu (fig. 5.5).

Trei probleme uzuale se pun în legătură cu circuitele :

a) fiind dat un circuit sub formă de diagramă, să se scrie funcția de transmisie a acesteia (analiză) ;

b) fiind dată o tablă de adevăr, să se scrie funcția și să se alcătuiască diagrama ;

c) să se înlocuiască un circuit printr-unul mai simplu, folosind procedeele de simplificare a funcțiilor de transmisie (minimizare).

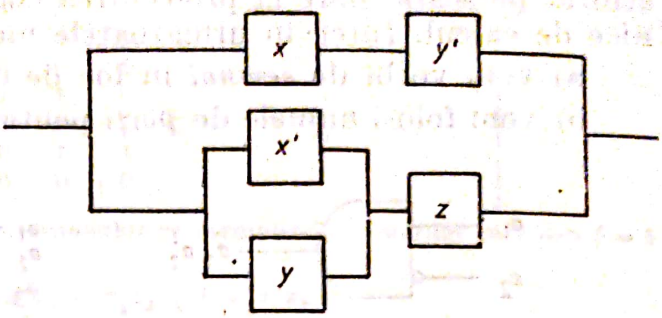


Fig. 5.4

Definiție. Fie $\langle X_n, +, \cdot \rangle$ o algebră booleană.

a) Elementele $x_1, \ldots, x_n \in X_n$ se numesc *contacte*.

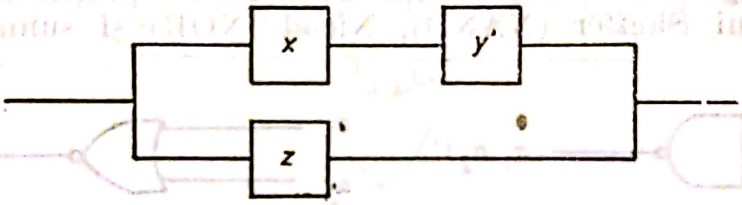


Fig. 5.5

b) Polinoamele booleene p aparținînd lui P_n se numesc *circuite cu contacte*.

c) x' este *contactul complementar* al lui x.

d) $x_i x_j$ este *legătura în serie* a lui x_i și x_j.

e) $x_i + x_j$ este *legătura în paralel* a lui x_i și x_j.

f) Funcția $\bar{p} \in P_n(B)$, corespunzătoare unui circuit, se numește *funcție de contact* (sau *transmisie*).

g) $\bar{p}(a_1, \ldots, a_n)$ este valoarea circuitului pentru $a_1, \ldots, a_n \in B$, iar a_i se numesc *variabile de intrare*.

În exemplul luat inițial am luat drept contact un comutator electric. Alte contacte bipoziționale sînt contactele mecanice, releele, fotocelulele,

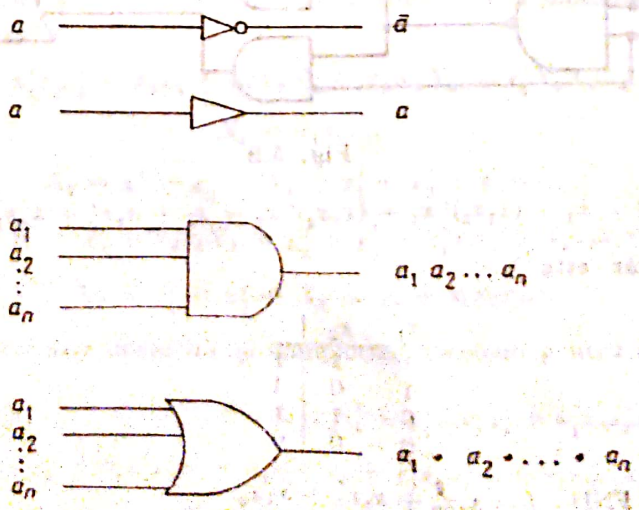


Fig. 5.6

Scanned with OKEN Scanner

tranzistorii. Cele două stări depind de natura contactelor: conductor/neconductor, pozitiv magnetizat/negativ magnetizat, încărcat/descărcat etc. Instrumentul algebric folosit pentru studiul circuitelor electrice este utilizat pe scară mare la proiectarea componentelor din mașinile electronice de calcul. Intervin următoarele modificări de vocabular:

a) vom vorbi de *semnal* în loc de curent electric,

b) vom folosi numele de *porți* pentru configurații de contacte.

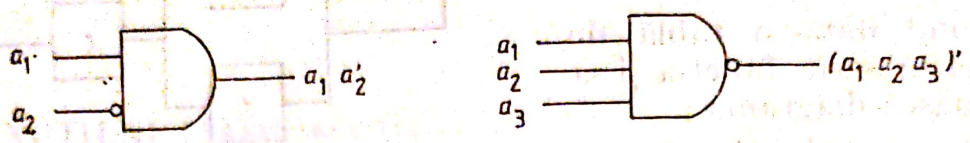


Fig. 5.7

Ca reprezentare grafică, în loc de blocuri apar simboluri reprezentînd porțile elementare după cum urmează din fig. 5.6: negația, identitatea, produsul (ȘI), suma (SAU). Un cerculeț la intrarea sau ieșirea dintr-o poartă înseamnă transformarea variabilei (sau rezultatului) în complementul său (fig. 5.7). O importanță deosebită au porțile care corespund conectorilor lui Sheffer (NAND), Nicod (NOR) și suma disjunctivă (fig. 5.8).

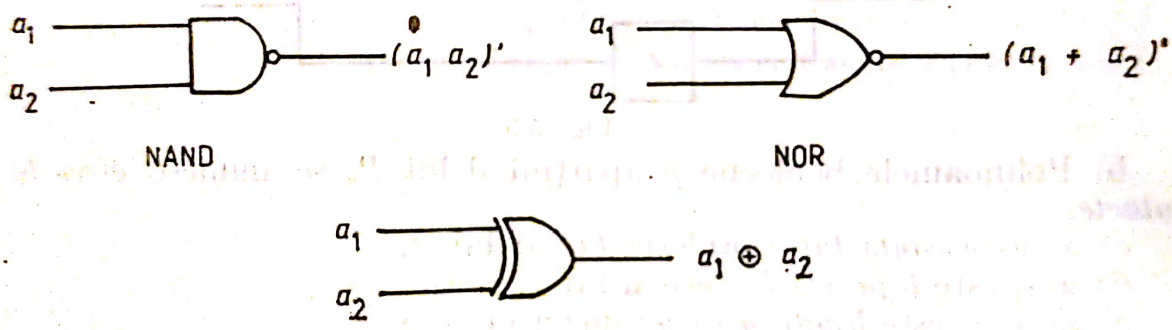


Fig. 5.8

Exemplu. *Analiză.* Să se scrie funcția corespunzătoare circuitului din fig. 5.9.

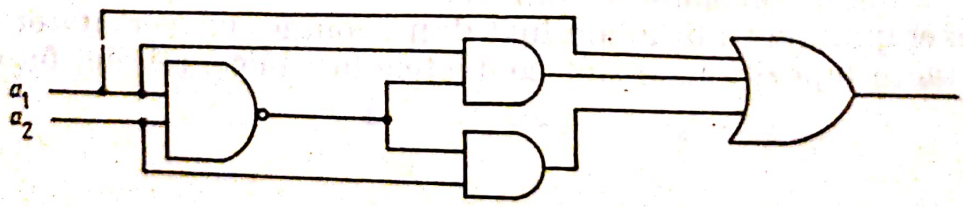


Fig. 5.9

$$f = x_1 + (x_1x_2)'\, x_1 + (x_1x_2)'\, x_2 = x_1 + x_1x_2' + x_2x_1'.$$

Tabla de adevăr este

x_1	x_2	
1	1	1
1	0	1
0	1	1
0	0	0

căreia îi corespunde FND: $f = x_1x_2 + x_1x_2' + x_1'x_2$.
Ambele forme sînt echivalente cu $f = x_1 + x_2$.

Scanned with OKEN Scanner

Sinteză. Să se reprezinte diagrama circuitului pentru care funcția de contact este definită de tabla

a_1	a_2	a_3	f
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

Forma FND a funcției este suma minitermenilor ce corespund evaluărilor care dau $f = 1$ în tablă:

$$f = x_1x_2x_3' + x_1x_2'x_3' + x_1'x_2x_3' + x_1'x_2'x_3.$$

Circuitul corespunzător este dat în fig. 5.10.

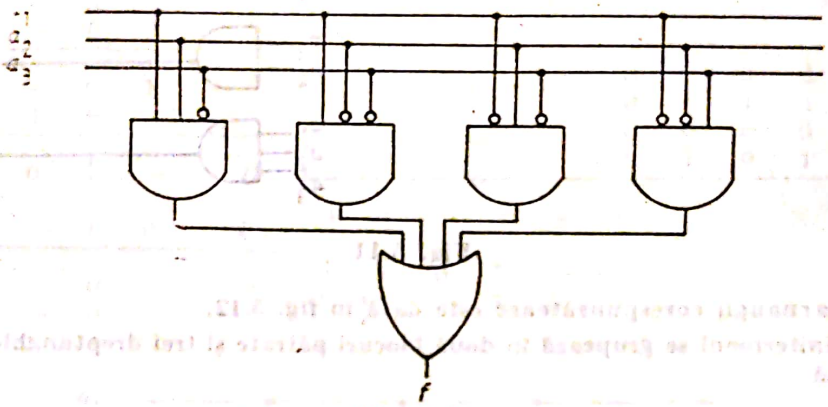


Fig. 5.10

5.2. MINIMIZARE

Înlocuirea unui circuit prin altul mai simplu, avînd aceeași funcție de contact, este o problemă importantă și frecventă. Interesul este dictat de economia de contacte (diode, tranzistori) și deci și de spațiu. Procedeul constă în găsirea *formei minimale* a funcției, recurgînd la unul din algoritmii cunoscuți (Karnaugh sau Quine-McCluskey).

Exemple. 1) Se cere analiza și simplificarea circuitului din 5.11, folosind harta Karnaugh. Pentru a ușura scrierea funcției, însemnăm ieșirile în cîteva puncte intermediare X_1, X_2, X_3, X_4 și astfel f devine

$$f = x_1'x_2'x_3' + x_2x_4 + x_1X_2X_4 + x_2X_3X_4 + x_4'X_2X_3X_4,$$

unde

$$X_1 = x_1x_2x_3,$$

$$X_2 = x_1' + x_3' + X_1 = x_1' + x_3' + x_1x_2x_3,$$

$$X_3 = x_2' + x_3' + x_3x_4X_2 = x_2' + x_3' + x_1'x_3x_4 + x_1x_2x_3x_4,$$

$$X_4 = x_1' + x_2' + X_1 = x_1' + x_1x_2x_3.$$

Înlocuind în f, efectuînd operațiile și completînd termenii pentru a primi forma FND, obținem

$$f = x_1'x_2'x_3'x_4' + x_1'x_2'x_3'x_4 + x_1'x_2'x_3x_4' + x_1'x_2x_3'x_4' + x_1'x_2x_3'x_4 +$$
$$+ x_1'x_2x_3x_4 + x_1x_2'x_3'x_4' + x_1x_2'x_3'x_4 + x_1x_2x_3'x_4 + x_1x_2x_3x_4' +$$
$$+ x_1x_2x_3x_4 = \Sigma(0, 1, 2, 4, 5, 7, 8, 9, 13, 14, 15).$$

Scanned with OKEN Scanner

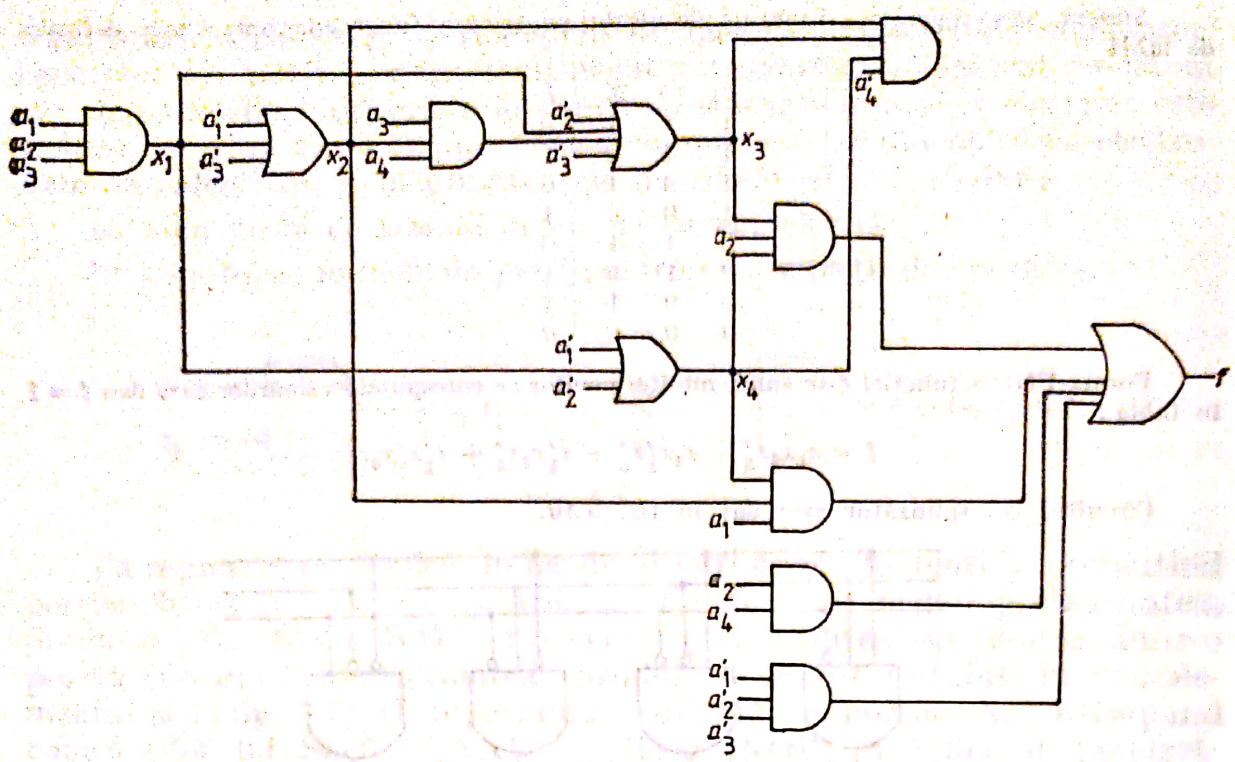


Fig. 5.11

Harta Karnaugh corespunzătoare este dată în fig. 5.12.

Cei 11 minitermeni se grupează în două blocuri pătrate și trei dreptunghiuri, care conduc la forma redusă:

$$f = x_2x_4 + x_2'x_3' + x_4'x_1'x_2' + x_3'x_4'x_1' + x_1x_2x_3$$

ceea ce revine la construirea unui circuit cu 6 porți în loc de 12.

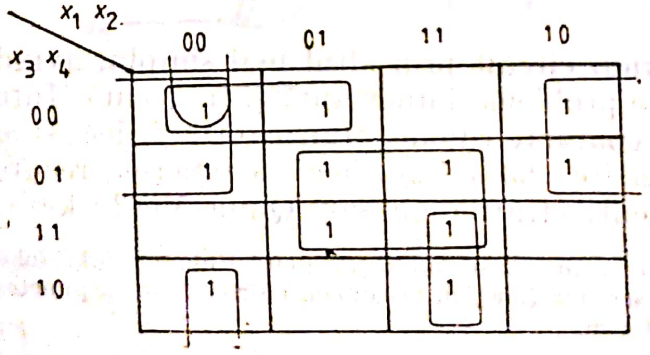


Fig. 5.12

2) Să se proiecteze circuitul minimal a cărui funcție de contact devine 1 pentru intrările (0, 0, 0, 0), (0, 0, 0, 1), (0, 1, 0, 0), (0, 1, 0, 1), (1, 0, 1, 0), (1, 0, 1, 1) și (1, 1, 1, 0).

Funcția de transfer este

$$f = x_1'x_2'x_3'x_4' + x_1'x_2'x_3'x_4 + x_1'x_2x_3'x_4' + x_1'x_2x_3'x_4 +$$

$$+ x_1x_2'x_3x_4' + x_1x_2'x_3x_4 + x_1x_2x_3x_4'.$$

Circuitul corespunzător ar necesita 8 porți.

Recurgem la algoritmul Quine-McCluskey (tabelul 5.1). În consecință, $A + B + C$ este forma minimală și

$$f = x_1'x_3' + x_1x_2'x_3 + x_1x_3x_4',$$

Scanned with OKEN Scanner

ceea ce permite proiectarea unui circuit cu un număr mai mic de porți (fig. 5.13).

Tabelul 5.1

	0	0	0	0	1
	0	0	0	1	2
	0	1	0	0	3
	0	1	0	1	4
	1	0	1	0	5
	1	0	1	1	6
	1	1	1	0	7
1,2	0	0	0	—	
1,3	0	—	0	0	
2,4	0	—	0	1	
3,4	0	1	0	—	
5,6	1	0	1	—	*B*
5,7	1	—	1	0	*C*
1,2 3,4	0	—	0	—	*A*
1,3 2,4	0	—	0	—	

	0	0	0	0	1	1	1
	0	0	1	1	0	0	1
	0	0	0	0	1	1	1
	0	1	0	1	0	1	0
A	0	—	0	—	×	×	×	×			
B	1	0	1	—					×	×	
C	1	—	1	0					×		×

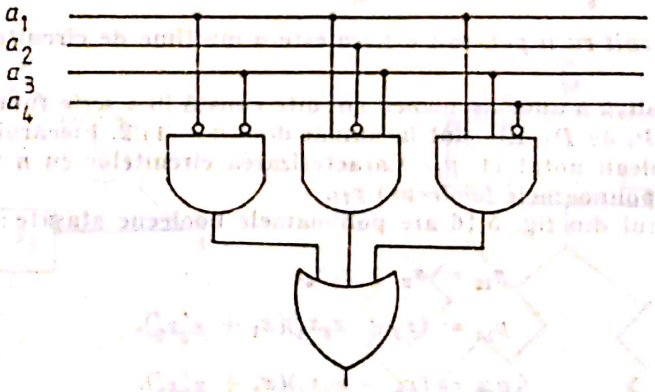


Fig. 5.13

5.3. PUNȚI ȘI CIRCUITE CU *n* POLI

Circuitele electrice care nu se compun din circuite în serie și în paralel se numesc *punți*. În cazul unui circuit punte, căutăm toate drumurile posibile și scriem funcția ca o sumă a acestor drumuri. Vom arăta că și acestei funcții îi va corespunde un circuit format din circuite în serie și în paralel. Revenim la scrierea cu contacte reprezentate de blocuri.

Exemplu. În fig. 5.14 curentul trece în următoarele condiții care corespund celor patru itinerarii posibile:

x_1 și x_2 închise; x_2 și x_4 închise;

x_1, x_3 și x_4 închise; x_2, x_3 și x_4 închise.

Ca urmare, $f = x_1x_2 + x_2x_4 + x_1x_3x_4 + x_2x_3$.

O formă mai simplă este

$$f = x_1x_3x_4 + x_2(x_1 + x_3 + x_4).$$

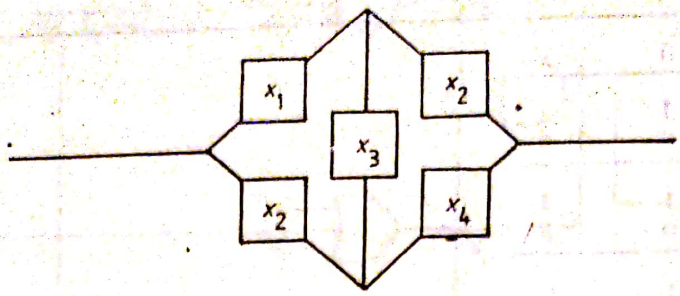


Fig. 5.14

Cele două circuite în serie și în paralel sînt date de fig. 5.15.

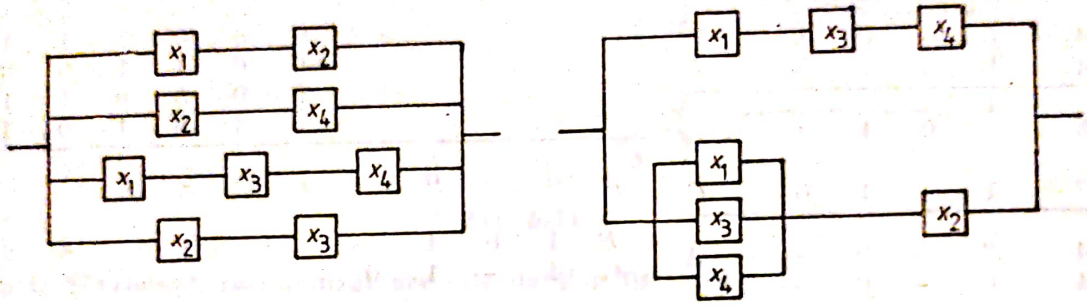


Fig. 5.15

Definiție. Un *circuit cu n poli* sau o *rețea* este o mulțime de circuite, avînd n puncte P_i, numite *poli*.

Procedeul de analiză a unor asemenea circuite constă în a scrie funcțiile corespunzătoare circuitelor care leagă P_i de P_j. Ele sînt în număr de $n(n-1)/2$. Fiecărui circuit îi corespunde astfel un polinom boolean notat cu p_{ij}. Caracterizarea circuitelor cu n poli revine la scrierea funcțiilor de contact (polinoamele booleene) p_{ij}.

Exemplu. Circuitul din fig. 5.16 are polinoamele booleene atașate:

$$p_{12} = x_2 + x_3x_4,$$

$$p_{13} = (x_3 + x_2x_4)(x_1 + x_2'x_3'),$$

$$p_{23} = (x_4 + x_2x_3)(x_1 + x_2'x_3').$$

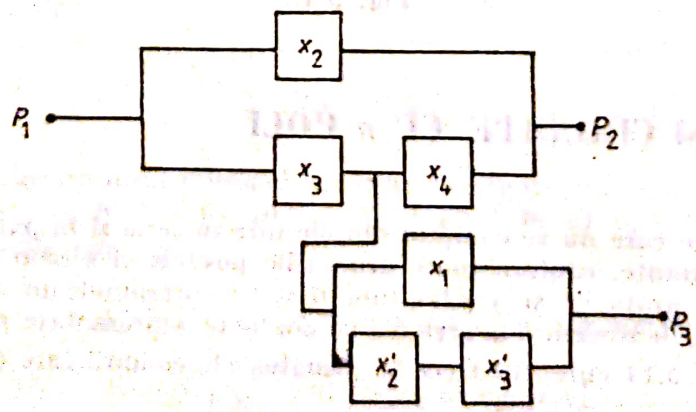


Fig. 5.16

Circuitele cu n poli echivalente au aceleași p_{ij}.

Circuitul stea definit ca un circuit cu 3 poli, în care cele trei circuite cu cîte doi poli au un punct care nu este pol, este echivalent cu *circuitul triunghi* în care nu există alte puncte de conexiune în afara polilor (fig. 5.17).

Scanned with OKEN Scanner

Următoarele două transformări permit trecerea de la o stea la un triunghi echivalent şi invers (fig. 5.18). Calculînd p_{ij} şi q_{ij} a celor două perechi de circuite, demonstrăm echivalenţa. Astfel, în fig. 5.18, b, avem

$$p_{13} = x_3 + x_1x_2, \qquad q_{13} = (x_1 + x_3)(x_3 + x_2) = x_3 + x_1x_2.$$

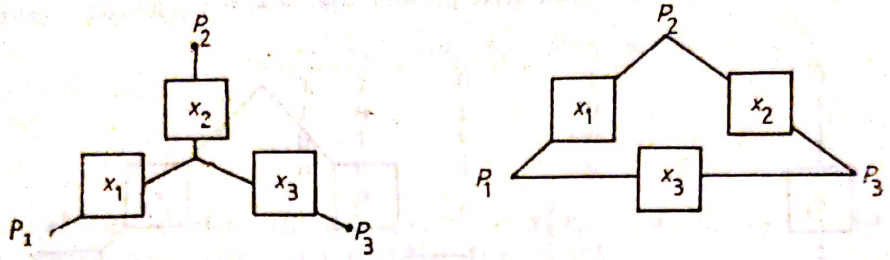
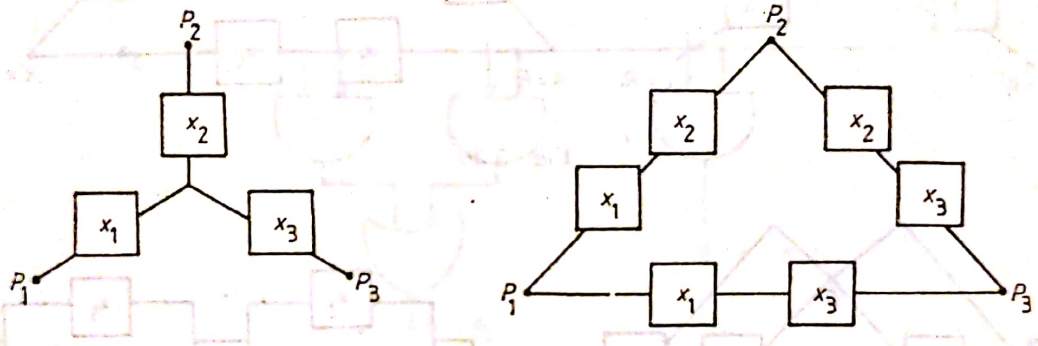


Fig. 5.17



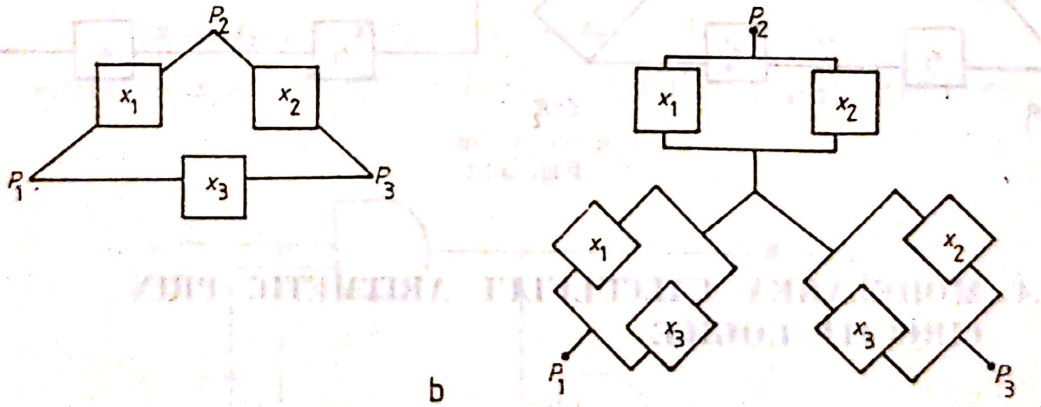


Fig. 5.18

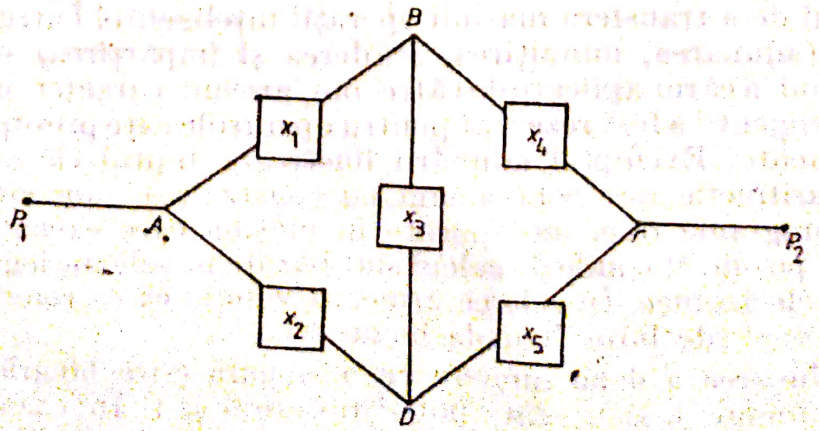


Fig. 5.19

Ne putem folosi de aceste transformări pentru a reduce un circuit punte la forma serie-paralel.

Fie circuitul din fig. 5.19. Îl reprezentăm sub formă de stea, omiţînd pe x_2 şi x_5, şi îl transformăm într-un triunghi (fig. 5.20). Punem la loc pe x_2 (între A şi D) şi pe x_5 (între C şi D) şi îl rescriem sub formă de circuit serie-paralel (fig. 5.21).

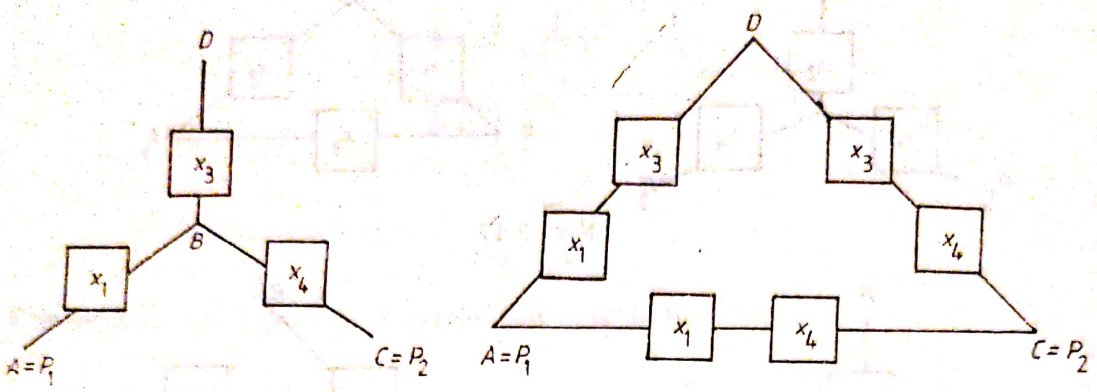


Fig. 5.20

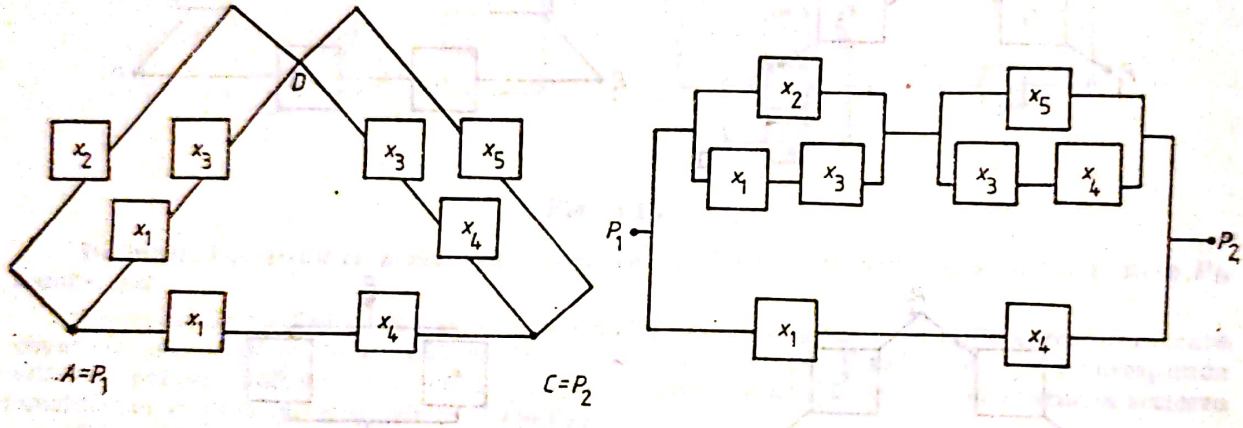


Fig. 5.21

5.4. MODELAREA CALCULULUI ARITMETIC PRIN CIRCUITE LOGICE

Automatizarea calculului aritmetic a fost una din primele realizări a procesului de a transfera maşinii operaţii inteligente. Întrucît operaţiile aritmetice (adunarea, înmulţirea, scăderea şi împărţirea) sînt efectuate prin algoritmi, a căror aplicare de către om are un caracter mecanic, atributul („inteligent" a fost rezervat pentru operaţiile care presupun programe şi raţionamente. Exemplul adunării ilustrează faptul că automatizarea calculului aritmetic este echivalentă cu construirea unor circuite logice, care sînt programe fixe, încorporate în maşina care execută operaţiile, substituind pe om. Maşinile de calcul sînt bazate pe scheme logice mecanice, electrice şi electronice. În cele ce urmează vom avea ca referinţă calculatorul electronic (de birou sau de buzunar).

a) **Adunarea a două numere cu o singură cifră binară.** Vom nota cele două numere a_1 şi a_2, care pot fi fiecare 0 şi 1. În cazul în care a_1 şi a_2 sînt egale cu 1, suma lor va fi 2, ceea ce în scrierea binară reclamă

două poziții 10. Se va efectua „*scrie* 0, *mută* 1" ca și în algoritmul adunării zecimale care este o nouă funcție logică pe care o vom nota $m(a_1, a_2)$ și care va trebui luată în considerare alături de rezultatele primei poziții, notate $s(a_1, a_2)$. Operația are următoarea tablă a valorilor celor două funcții:

a_1	a_2	$\bar{s}(a_1, a_2)$	$\bar{m}(a_1, a_2)$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

Din tablou deducem $s = x_1x_2' + x_1'x_2 = x_1 \oplus x_2$ și $m = x_1x_2$, iar circuitul corespunzător este dat în fig. 5.22.

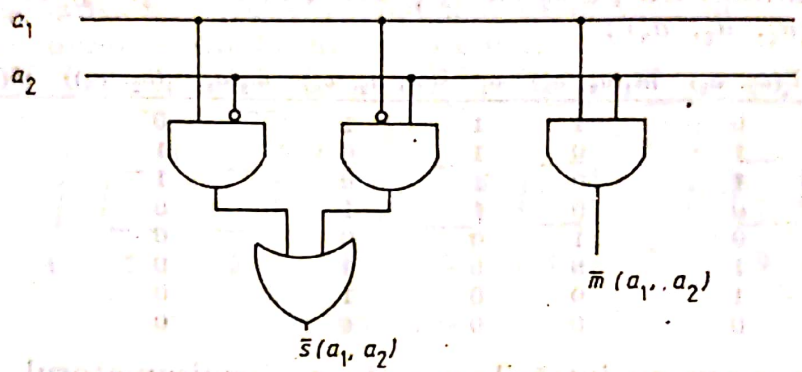


Fig. 5.22

Putem simplifica acest circuit și anume căutăm să exprimăm una din cele două funcții în raport cu cealaltă:

$$s = x_1x_2' + x_1'x_2 = (x_1' + x_2)' + (x_1 + x_2')' =$$
$$= ((x_1' + x_2)(x_1 + x_2'))' = (x_1x_2 + x_1'x_2')' = (x_1x_2)'\,(x_1'x_2')' =$$
$$= m'(x_1 + x_2).$$

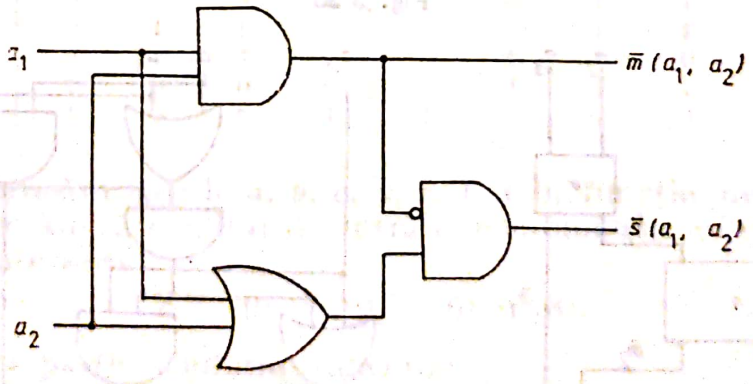


Fig. 5.23

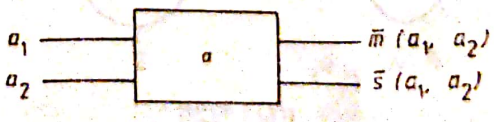


Fig. 5.24

Circuitul se poate construi astfel cu trei porți (fig. 5.23). Acest circuit îl numim *semisumator*, și este reprezentat în fig. 5.24.

Scanned with OKEN Scanner

b) **Adunarea a trei numere cu o singură cifră binară.** Vom întocmi o tablă a valorilor funcțiilor $\bar{s}(a_1, a_2, a_3)$ și $\overline{m}(a_1, a_2, a_3)$:

a_1	a_2	a_3	$\bar{s}(a_1, a_2, a_3)$	$\overline{m}(a_1, a_2, a_3)$
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

Vom descompune apoi adunarea celor trei numere în două adunări parțiale, adunînd întîi pe a_2 cu a_3 și rezultatul $\bar{s}_1(a_2, a_3)$ cu a_1 pentru a obține pe $\bar{s}(a_1, a_2, a_3)$:

a_2	a_3	$\bar{s}_1(a_2, a_3)$	$\overline{m}_1(a_2, a_3)$	a_1	$\bar{s}(a_1, a_2, a_3)$	$\overline{m}_2(a_1, s_1(a_2, a_3))$	$\overline{m}(a_1, a_2, a_3)$
1	1	0	1	1	1	0	1
1	0	1	0	1	0	1	1
0	1	1	0	1	0	1	1
0	0	0	0	1	1	0	0
1	1	0	1	0	0	0	1
1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0
0	0	0	0	0	0	0	0

Formăm acum cu intrările a_2 și a_3 semisumatorul cu ieșirile $\bar{s}_1(a_2,a_3)$ și $\overline{m}_1(a_2,a_3)$. Formăm al doilea semisumator cu intrările $\bar{s}_1(a_2, a_3)$ și a_1 și ieșirile $\bar{s}_1(a_1, a_2, a_3)$ și $\overline{m}_2(a_1, \bar{s}_1(a_2, a_3))$. Însumarea lui $\overline{m}_1(a_2, a_3)$ și $\overline{m}_2(a_1, \bar{s}_1(a_2. a_3))$ ne va da pe $\overline{m}(a_1, a_2, a_3)$.

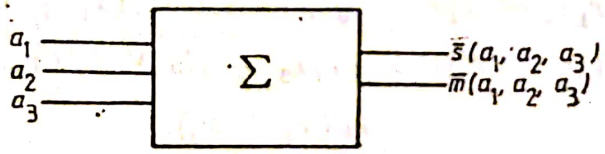


Fig. 5.25

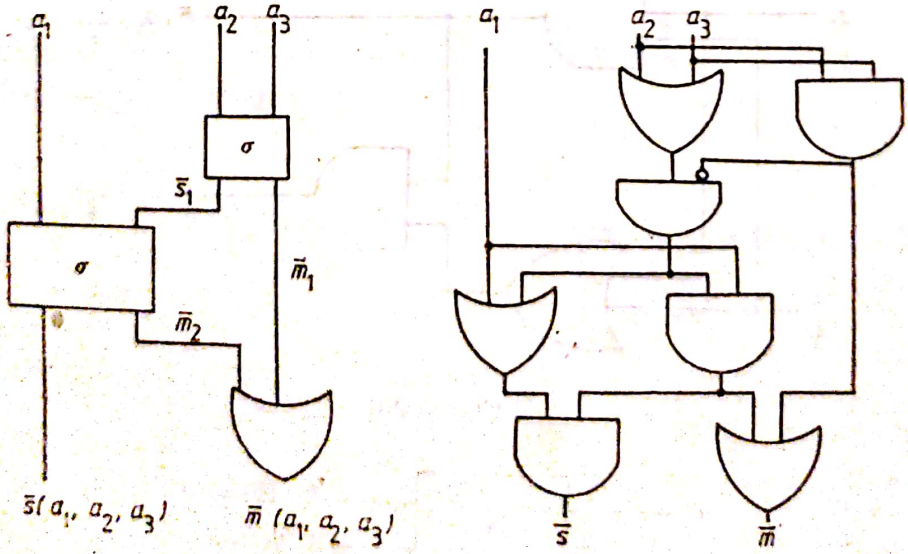


Fig. 5.26

Circuitul corespunzător va fi notat cu Σ și va avea trei intrări și două ieșiri (fig. 5.25). Structura lui va fi dată în raport cu σ sau cu explicitarea porților (fig. 5.26).

Scanned with OKEN Scanner

5.5. PROIECTAREA LOGICĂ A AFIȘAJULUI

Afișajul cifrelor zecimale la mașinile de calcul, la ceasurile electronice și la alte aparate se bazează pe următoarele principii : un ecran digital constînd din șapte segmente luminoase notate cu

a, b, c, d, e, f, g (fig. 5.27)

permite afișarea oricărei cifre (fig. 5.28).

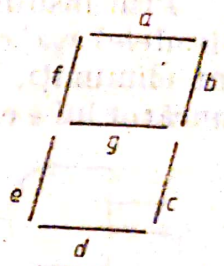


Fig. 5.27

Numerele zecimale sînt reprezentate de codul binar al decimalelor (CBD), necesitînd patru intrări. Problema constă în proiectarea unui circuit, care să transforme cifrele zecimale reprezentate binar la intrare printr-o ieșire, cuprinzînd o comandă către cele șapte segmente luminoase, astfel ca acționarea lor conform comenzii să le facă să corespundă cu numerele de la intrare (tabelul 5.2).

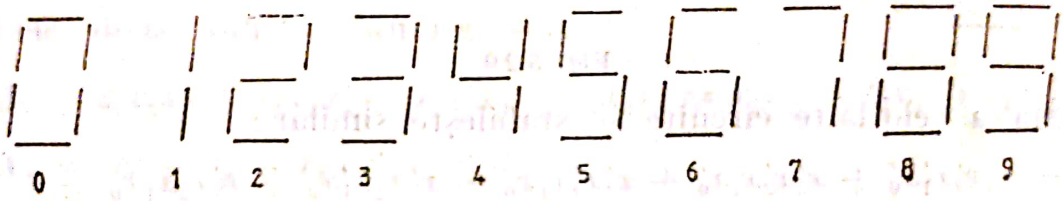


Fig. 5.28

Tabelul 5.2

Cifră zecimală	Intrare										
	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Ca să exprimăm ieșirile a, b, c, d, e, f, g în funcție de intrările x_3, x_2, x_1, x_0, observăm că rezultatul e intră în compunerea cifrelor 0, 2, 6, 8, ceea ce se poate scrie

$$e = (0) \vee (2) \vee (6) \vee (8).$$

Pe de altă parte, conform tabelului,

$$(0) = x_3' x_2' x_1' x_0' \qquad (6) = x_3' x_2 x_1 x_0'$$

$$(2) = x_3' x_2' x_1 x_0' \qquad (8) = x_3 x_2' x_1' x_0'$$

Astfel

$$e = x_3' x_2' x_1' x_0' + x_3' x_2' x_1 x_0' + x_3' x_2 x_1 x_0' + x_3 x_2' x_1' x_0'.$$

Dar e se poate aduce la forma minimală :

$$e = x_2' x_0' + x_1 x_0'.$$

Scanned with OKEN Scanner

Într-adevăr, $x_2'x_0' = x_2'x_0'(x_3'x_1' + x_3'x_1 + x_3x_1' + x_3x_1) = \underline{x_3'x_2'x_1'x_0'} +$ $+ \underline{x_3'x_2'x_1x_0'} + \underline{x_3x_2'x_1'x_0'} + x_3x_2'x_1x_0'$, iar $x_1x_0' = x_1x_0'(x_3'x_2' + x_3'x_2 + x_3x_2' +$ $+ x_3x_2) = x_3'x_2'x_1x_0' + x_3'x_2x_1x_0' + x_3x_2'x_1x_0' + x_3x_2x_1x_0'$.

Prin însumare obținem e (termenii subliniați). Pseudocodul (pseudotetradele) $x_3x_2'x_1x_0'$, $x_3x_2x_1x_0'$, care nu au corespondent la intrările tabloului, sînt eliminați. Mai departe putem scrie $e = x_0'(x_2' + x_1)$. Circuitul corespunzător lui e este dat de fig. 5.29.

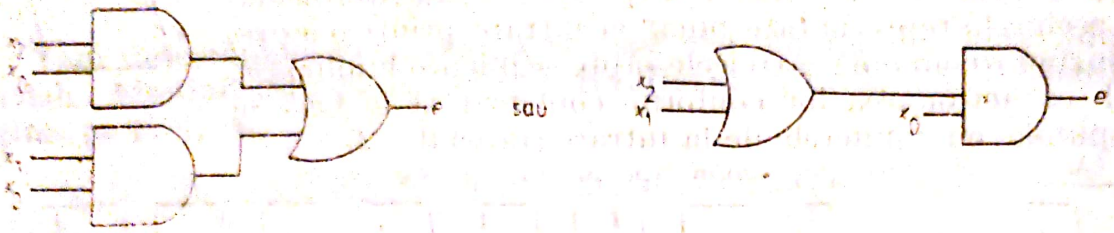


Fig. 5.29

Ecuația celorlalte circuite se stabilește similar:

$$a = x_3'x_2'x_1'x_0' + x_3'x_2'x_1x_0' + x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0 + x_3'x_2x_1x_0 +$$
$$+ x_3x_2'x_1'x_0' + x_3x_2'x_1x_0,$$

$$b = x_3'x_2'x_1'x_0' + x_3'x_2'x_1'x_0 + x_3'x_2'x_1x_0' + x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0' +$$
$$+ x_3'x_2x_1x_0 + x_3x_2'x_1'x_0' + x_3x_2'x_1'x_0,$$

$$c = x_3'x_2'x_1'x_0' + x_3'x_2'x_1'x_0 + x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0' + x_3'x_2x_1'x_0 +$$
$$+ x_3'x_2x_1x_0' + x_3'x_2x_1x_0 + x_3x_2'x_1'x_0' + x_3x_2'x_1'x_0,$$

$$d = x_3'x_2'x_1'x_0' + x_3'x_2'x_1x_0' + x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0 + x_3'x_2x_1x_0' +$$
$$+ x_3x_2'x_1'x_0' + x_3x_2'x_1'x_0,$$

$$f = x_3'x_2'x_1'x_0' + x_3'x_2x_1'x_0' + x_3'x_2x_1'x_0 + x_3'x_2x_1x_0' + x_3x_2'x_1'x_0' +$$
$$+ x_3x_2'x_1'x_0,$$

$$g = x_3'x_2'x_1'x_0' + x_3'x_2'x_1x_0 + x_3'x_2x_1'x_0' + x_3'x_2x_1'x_0 + x_3'x_2x_1x_0' +$$
$$+ x_3x_2'x_1'x_0' + x_3x_2'x_1'x_0.$$

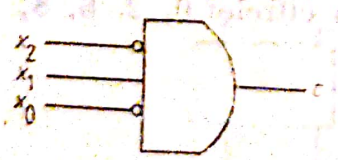


Fig. 5.30

Lăsăm ca exercițiu simplificarea acestor șase funcții, reprezentarea circuitelor corespunzătoare și asamblarea lor pentru rezolvarea problemei (cel mai simplu dintre ele este circuitul corespunzător segmentului) din fig. 5.30.

5.6. ALTE APLICAȚII

Următoarele exemple sînt alese din domenii diferite (energie electrică, funcționarea ascensoarelor, industria chimică).

1) **Sistem de alarmă.** Un motor este alimentat de trei generatoare. Cînd un generator se oprește, un monitor închide un circuit. Proiectăm

un sistem de alarmă care constă din aprinderea unui bec cînd un generator din două se oprește și un sistem de alarmă acustică (sirena) cînd două sau trei generatoare nu funcționează. Notăm cu a_1, a_2, a_3 semnalul (intrarea) asupra funcționării fiecărui generator. Cînd $a_i = 1$, generatorul i este oprit. Notăm cu $A_1(a_1, a_2, a_3)$ și $A_2(a_1, a_2, a_3)$ cele două sisteme de alarmă (becul și sirena). Dacă $A_j = 1$, înseamnă că sistemul j este acționat. Tabla celor două funcții traduce criteriile fixate:

a_1	a_2	a_3	A_1	A_2
1	1	1	1	0
1	1	0	1	1
1	0	1	1	1
1	0	0	0	1
0	1	1	1	1
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

Cele două funcții căutate sînt:

$$A_1 = x_1x_2x_3 + x_1x_2x_3' + x_1x_2'x_3 + x_1'x_2x_3 = x_1x_2 + x_2x_3 + x_1x_3,$$

$$A_2 = (x_1 + x_2 + x_3)(x_1' + x_2' + x_3'),$$

iar răspunsul la problemă va fi dat de circuitul din fig. 5.31.

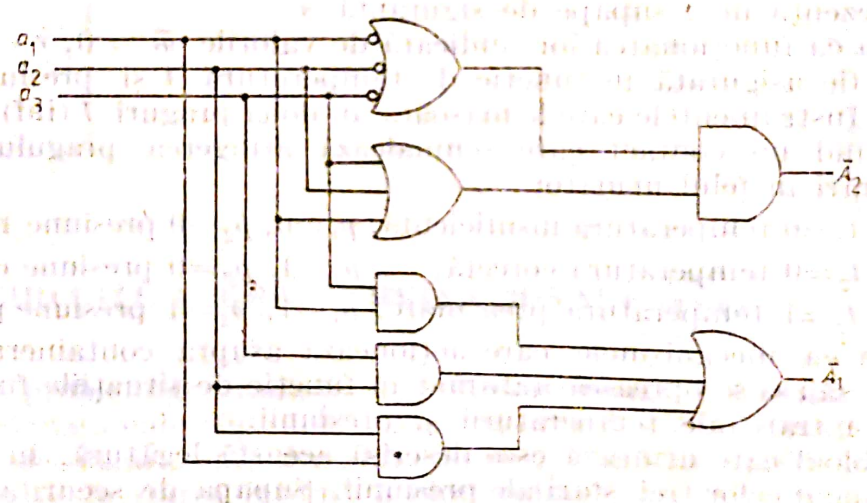


Fig. 5.31

2) **Ascensorul.** Pentru simplificare considerăm cazul a trei etaje (e_1, e_2, e_3), unde ascensorul se poate afla la un moment dat și a trei apeluri (a_1, a_2, a_3) ce pot fi făcute la cele trei etaje. La fiecare combinație ascensorul poate să meargă într-una din două direcții: sus (S) sau jos (J). Din cele $2^6 = 64$ intrări posibile, reținem pe cele corespunzînd situațiilor cînd cele trei butoane de apel sînt apăsate în 2^3 combinații (unul din trei, cîte două simultan, toate trei), pentru cele trei etaje. Criteriile de funcționare sînt următoarele: ascensorul merge unde este chemat; nu pornește dacă nu este apel sau dacă apelul vine de la etajul unde el se află. Cînd este la etajul 2 și este chemat simultan la etajele 1 și 3, pornește în direcția jos. Notăm cu $a_i = 1$ prezența apelului la etajul i, iar $e_j = 1$ înseamnă că ascensorul se află la etajul j. Convenim că $\bar{S} = 1$ are semnificația pornirii în sus, iar $\bar{J} = 1$ în jos.

Scanned with OKEN Scanner

Reprezentăm tabla

a_1	a_2	a_3	$e_1(e_2=e_3=0)$	$\bar{S}$	$\bar{J}$	$e_2(e_1=e_3=0)$	$\bar{S}$	$\bar{J}$	$e_3(e_1=e_2=0)$	$\bar{S}$	$\bar{J}$
1	1	1	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0	1	0	1
1	0	1	1	0	0	1	0	1	1	0	0
1	0	0	1	0	0	1	0	1	1	0	1
0	1	1	1	1	0	1	0	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	0	1	1	1	0	1	1	0	1	0	0
0	0	0	1	0	0	1	0	0	1	0	0

Cele două funcții S și J sînt conform tablei :

$$S = a_1'a_2a_3e_1e_2'e_3' + a_1'a_2a_3'e_1e_2'e_3' + a_1'a_2'a_3e_1e_2'e_3' + a_1'a_2'a_3e_1'e_2e_3' =$$

$$= a_1'a_2a_3'e_1e_2'e_3' + a_1'a_3e_1e_2'e_3' + a_1'a_2'a_3e_1'e_2e_3',$$

$$J = a_1a_2'a_3e_1'e_2e_3' + a_1a_2'a_3'e_1'e_2e_3' + a_1a_2a_3'e_1'e_2'e_3 + a_1a_2'a_3'e_1'e_2'e_3 +$$

$$+ a_1'a_2a_3'e_1'e_2'e_3 = a_1a_2'e_1'e_2e_3' + a_1a_3'e_1'e_2'e_3 + a_1'a_2a_3'e_1'e_2'e_3.$$

c) **Controlul reacției chimice.** Într-un container, producerea reacției chimice este rezultatul a patru factori :

a) amestecul, acționat de un motor m ;

b) introducerea apei de răcire r ;

c) acționarea unui mecanism de încălzire c ;

d) prezența unei supape de siguranță s.

Dorim ca funcționarea lor (indicată de valorile $\bar{m} = 0$, $\bar{r} = 1$, $\bar{c} = 1$, $\bar{s} = 1$) să fie asigurată în funcție de temperatura t și presiunea p din container. Instrumentele care le măsoară au două praguri *1* (inf) și *2* (sup), fiecare avînd un contact care semnalează atingerea pragului. Notăm aceste intrări în felul următor :

$t_1=0$, $t_2=0$ temperatura insuficientă, $p_1=0$, $p_2=0$ presiune mică,

$t_1=1$, $t_2=0$ temperatura corectă, $p_1=1$, $p_2=0$ presiune corectă,

$t_1=1$, $t_2=1$ temperatura prea mare, $p_1=1$, $p_2=1$ presiune prea mare.

Dorim ca mecanismele care acționează asupra containerului să se declanșeze sau să se oprească automat în funcție de situațiile furnizate de datele de intrare ale temperaturii și presiunii.

În tabloul care urmează este descrisă această legătură, în trei etape care corespund celor trei stări ale presiunii. Supapa de securitate intră în acțiune atunci cînd atît temperatura cît și presiunea sînt critice.

$\bar{t}_1$	$\bar{t}_2$	$\bar{p}_1$	$\bar{p}_2$	$\bar{c}$	$\bar{r}$	$\bar{m}$	$\bar{s}$
0	0	0	0	1	0	1	0
1	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0
0	0	1	0	1	0	0	0
1	0	1	0	0	0	0	0
1	1	1	0	0	1	1	0
0	0	1	1	0	0	1	0
1	0	1	1	0	1	1	0
1	1	1	1	0	1	1	1

Funcțiile corespunzătoare vor fi :

$$c = t_1't_2'p_1'p_2' + t_1t_2'p_1'p_2' + t_1't_2'p_1p_2' = t_2'p_2'(t_1' + p_1'),$$

$$r = t_1t_2p_1p_2' + t_1t_2'p_1p_2 + t_1t_2p_1p_2 = t_1p_2(t_2 + p_2),$$

Scanned with OKEN Scanner

$$m = t_1't_2'p_1'p_2' + t_1t_2p_1'p_2' + t_1t_2p_1p_2' + t_1't_2'p_1p_2 + t_1t_2'p_1p_2 + t_1t_2p_1p_2 =$$
$$= t_1't_2'p_1p_2 + t_1t_2p_1p_2 + t_1t_2p_2' + t_2'p_1p_2,$$
$$s = t_1t_2p_1p_2.$$

Soluția este dată de circuitul din fig. 5.32.

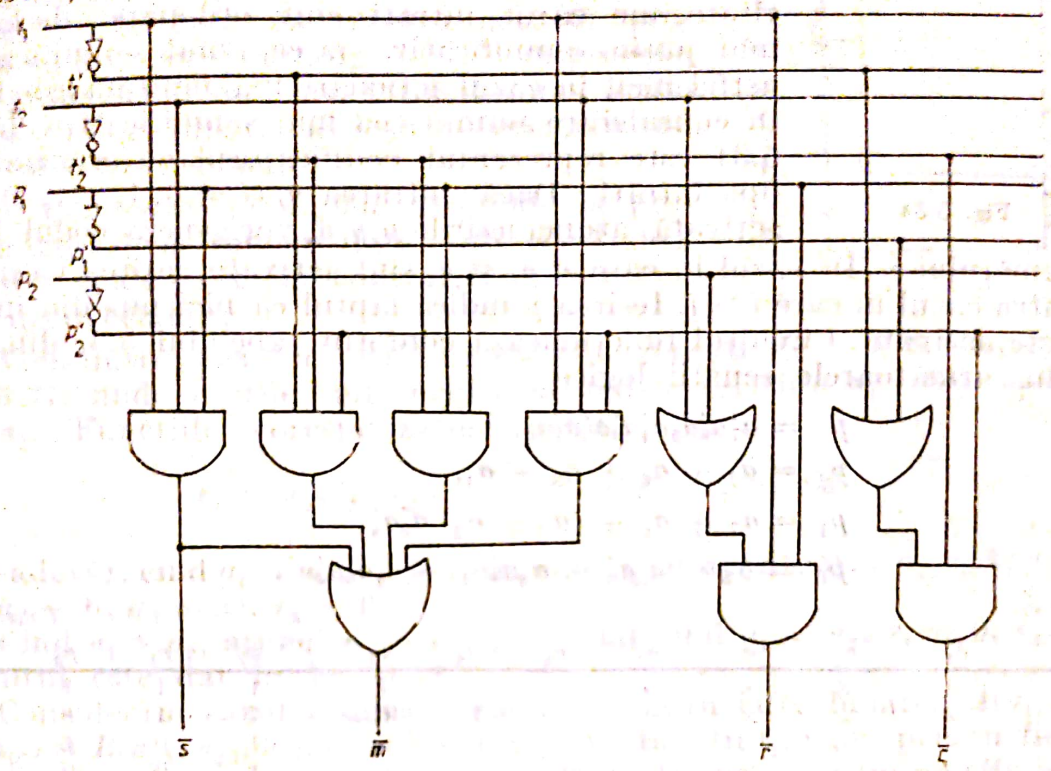


Fig. 5.32

5.7. CIRCUITE LOGICE COMBINAȚIONALE (CLC)

Toate problemele studiate mai înainte intră în familia circuitelor logice combinaționale (fig. 5.33), unde p_i sînt funcții depinzînd doar de variabilele de intrare, spre deosebire de circuitele secvențiale, la care ieșirile depind și de starea internă a circuitului.

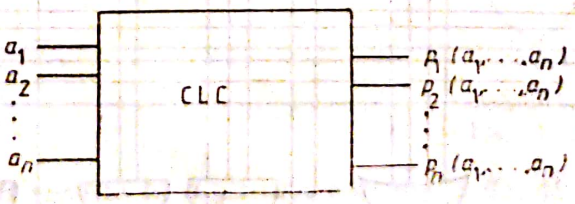


Fig. 5.33

Un CLC este astfel un ansamblu (de circuite corespunzînd unei probleme determinate), care leagă datele de la ieșire de datele de la intrare. Proiectarea în microelectronică a circuitelor logice combinaționale face parte integrantă din proiectarea circuitelor integrale digitale.

În construirea mașinilor electronice de calcul și a altor produse ale microelectronicii, anumite probleme intervin frecvent (codificări, decodificări, comparator, digital, sumatoare) și de aceea pachetele de circuite

Scanned with OKEN Scanner

logice care le corespund sînt proiectate și produse în module standardizate. Schițăm principiile care stau la baza celor mai curente:

a) *Codificatorul* rezolvă probleme în care unei intrări active îi corespunde un cuvînt binar de k biți. Pentru a se evita comportamente disfuncționale în cazul activității a două sau mai multe intrări, a fost definit codificatorul cu prioritate, ale cărui intrări sînt ordonate, de la cea mai puțin semnificativă la cea mai semnificativă, astfel incît în cazul acționării mai multora este luată în considerare numai cea mai semnificativă. În fig. 5.34 este reprezentat codificatorul cu prioritate cu opt intrări. Dacă intrarea a_i $(i = 0, 1, \ldots, 7)$ este activată, atunci ieșirile $p_2p_1p_0$ vor genera codul binar al numărului i. În cazul în care și a_i și a_j sînt activate, ieșirea va indica i pentru cazul în care $i > j$. Ieșirea p indică faptul că nici una din intrări nu este activată. Circuitul funcționează conform tabelului 5.3, din care rezultă următoarele ecuații logice:

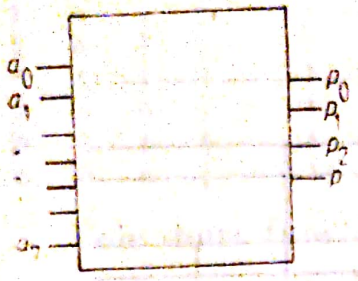


Fig. 5.34

$$p = a_7'a_6'a_5'a_4'a_3'a_2'a_1'a_0',$$
$$p_2 = a_7 + a_6 + a_5 + a_1,$$
$$p_1 = a_7 + a_6 + (a_3 + a_2)\,a_5'a_1',$$
$$p_0 = a_7 + a_5a_6' + a_3a_6'a_4' + a_1a_6'a_4'a_2',$$

Tabelul 5.3

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	p_2	p_1	p_0	p
×	×	×	×	×	×	×	1	1	1	1	0
×	×	×	×	×	×	1	0	1	1	0	0
×	×	×	×	×	1	0	0	1	0	1	0
×	×	×	×	1	0	0	0	1	0	0	0
×	×	×	1	0	0	0	0	0	1	1	0
×	×	1	0	0	0	0	0	0	1	0	0
×	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1

și schema asociată lor reprezentată în fig. 5.35.

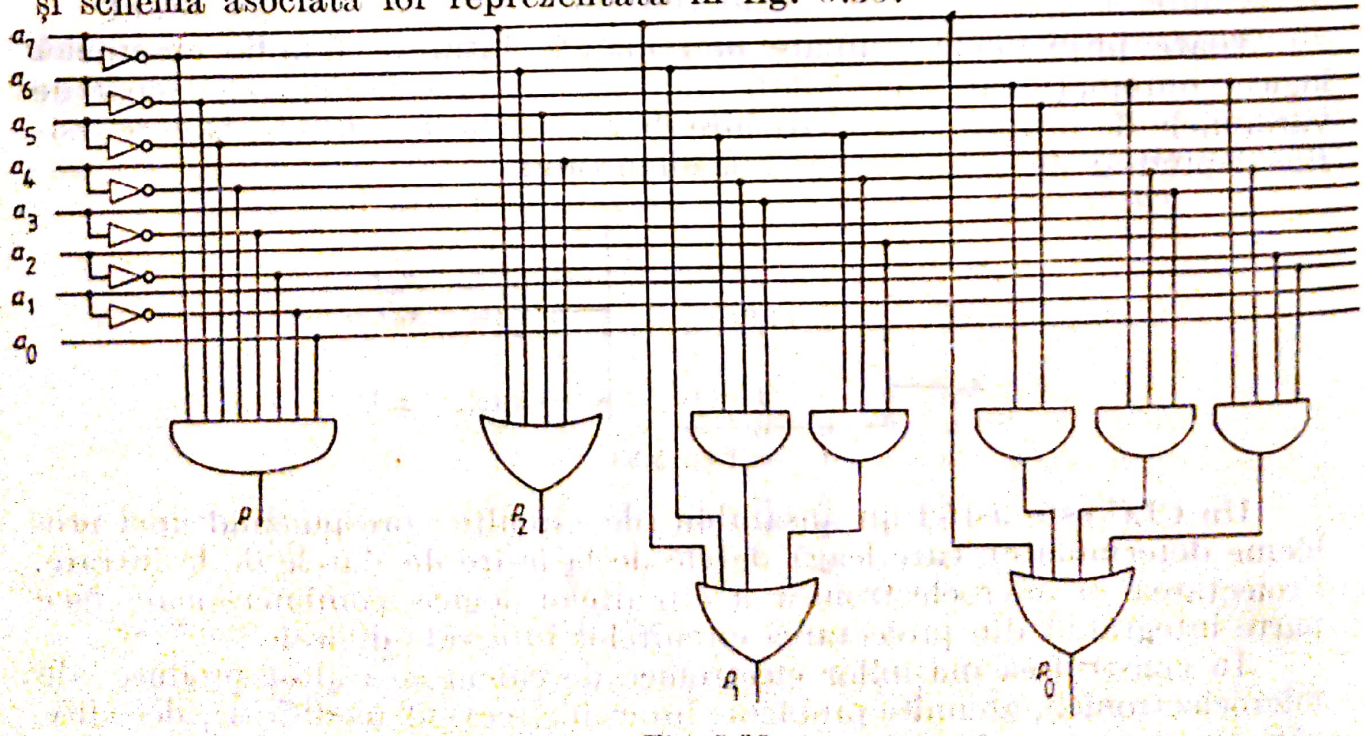


Fig. 5.35

b) *Decodificatorul* este un CLC care rezolvă problema inversă și anume, fiind dat la intrare codul (sub forma unui cuvînt binar de k biți), la ieșire se obține singurul cuvînt pe care îl codifică. Sînt produse codificate pentru diferite tipuri de coduri.

Pentru ilustrare vom pune problema decodificării trecerii de la scrierea binară la cea zecimală. CLC care va rezolva această problemă va fi de forma din fig. 5.36. Vom proceda ca în problema de mai sus, intrările fiind acum cele patru cifre binare, iar ieșirile corespunzînd celor 10 cifre zecimale.

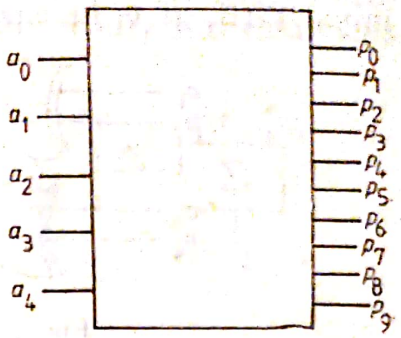


Fig. 5.36

c) *Comparatorul digital* răspunde la problema dacă două numere date (sub formă de k biți) sînt egale și, în caz contrar, care este mai mare decît celălalt.

Vom ilustra cazul a două numere reprezentate de o cifră binară prin fig. 5.37, unde ieșirile sînt menite să indice relațiile $a_1 < a_2$, $a_1 = a_2$, $a_1 > a_2$. Funcțiile corespunzătoare sînt:

$$r_1 = x_1x_2', \quad r_2 = x_1x_2 + x_1'x_2', \quad r_3 = x_1'x_2.$$

Într-adevăr, cînd $a_1 = a_2$, $\bar{r}_2 = 1$, iar cînd $a_1 \neq a_2$ (situația $a_1 = 1$, $a_2 = 0$ sau $a_1 = 0$, $a_2 = 1$), $\bar{r}_2 = 0$.

Cînd $a_1 > a_2$, atunci $\bar{r}_1 = 1$, $\bar{r}_3 = 0$, iar cînd $a_1 < a_2$, $\bar{r}_1 = 0$, $\bar{r}_3 = 0$. Circuitul este dat în fig. 5.38.

Considerăm cazul a două numere cu patru cifre binare, $A(a_{13}, a_{12}, a_{11}, a_{10})$ și $B(a_{23}, a_{22}, a_{21}, a_{20})$. Vom avea trei funcții r_1, r_2, r_3 pentru fiecare din ele, al doilea indice din r_{13}, r_{23}, r_{33} arată că ne ocupăm de cifrele a_{13}, a_{23} etc. Vom nota ca și mai înainte funcția R_1 pentru cazul $A > B$, R_2 pentru cazul $A = B$ și R_3 pentru $A < B$. Egalitatea celor două numere presupune egalitatea dintre cifrele binare corespunzătoare:

$$R_2 = r_{20}\, r_{21} r_{22}\, r_{23}$$

care se realizează prin legarea cu *și* a patru circuite de tip r_2.

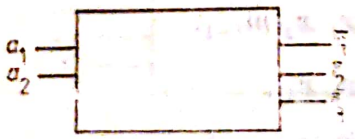


Fig. 5.37

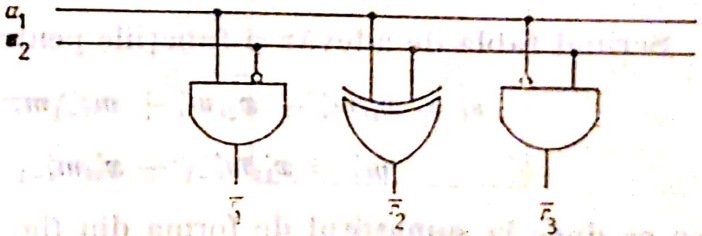


Fig. 5.38

Condiția $A > B$ revine la $a_{13} > a_{23}$ sau $a_{12} > a_{22}$ și $a_{13} = a_{23}$ sau $a_{11} > a_{21}$ și $a_{12} = a_{22}$ și $a_{13} = a_{23}$ sau $a_{10} > a_{20}$ și $a_{11} = a_{21}$ și $a_{12} = a_{22}$ și $a_{13} = a_{23}$, ceea ce se poate scrie $R_1 = r_{13} + r_{12}\, r_{23} + r_{11} r_{22} r_{23} + r_{10} r_{21} r_{22} r_{23}$.

La fel, $R_3 = r_{33} + r_{32} r_{23} + r_{31} r_{22} r_{23} + r_{30} r_{21} r_{22} r_{23}$.

Folosind relațiile care exprimă că numai una din situațiile $A > B$, $A = B$, $A < B$ este adevărată, avem $R_1 = R_2'R_3'$, $R_2 = R_1'\, R_3'$, $R_3 = R_1'R_2'$, ceea ce indică posibilitatea construirii circuitelor care corespund la două din aceste funcții, a treia obținîndu-se din ele cu ajutorul a două inversări și a unei porți *și*.

Scanned with OKEN Scanner

d) *Detectorul de paritate* răspunde la întrebarea dacă numărul variabilelor de la intrare egale cu 1 este par sau nu.

Se recurge la conectorul logic al sumei disjuncte a cărei tablă de adevăr indică 1, numai cînd variabilele sînt diferite (1 și 0), deci în cazul imparității, și 0 în cazul parității.

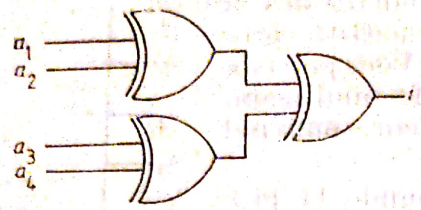


Fig. 5.39

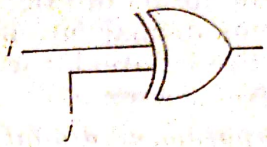


Fig. 5.40

Pentru patru variabile schema unui detector de imparitate este dată de fig. 5.39. Dacă adăugăm încă o poartă și o intrare j, obținem fig. 5.40, și detectorul servește ca detector de imparitate cînd $j = 0$ și ca detector de paritate pentru $j = 1$.

e) *Sumatorul* servește la efectuarea operațiilor aritmetice. Adunarea a două și trei numere cu o singură cifră binară a fost analizată în 5.4. Prin folosirea porții sumei disjunctive semisumatorul se simplifică (fig.5.41).

Cînd două numere se scriu cu n biți, adunarea cifrelor de același rang se realizează cu trei intrări, două corespunzînd celor două cifre și a treia mutării provenite de la adunarea cifrelor de rang imediat anterior. Circuitul este dat de sumatorul din fig. 5.42.

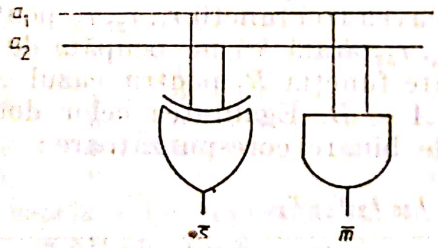


Fig. 5.41

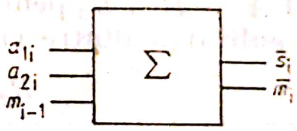


Fig. 5.42

Scriind tabla de adevăr și funcțiile pentru acest caz, obținem

$$s_i = x_{1i}m'_i + x_{2i}m'_i + m_{i-1}m_i + x_{1i}x_{2i}m_{i-1},$$

$$m'_i = x'_{1i}m'_{i-1} + x'_{2i}m'_{i-1} + x'_{1i}x'_{2i},$$

ceea ce duce la sumatorul de forma din fig. 5.43.

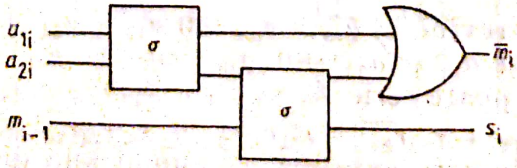


Fig. 5.43

Adunarea a două numere cu n biți se poate realiza în paralel, conform schemei din fig. 5.44.

f) *Multiplexorul* îndeplinește funcția de a transmite datele de la una din cele m intrări la o unică ieșire. Selecția intrării se realizează printr-un cod de selecție de n biți, astfel ca $m = 2^n$. Diferitele tipuri de multiplexoare se definesc prin m și n, la care se adaugă caracteristica dacă ieșirea este completată, necompletată (sau ambele). Aplicațiile multiplexorului sînt

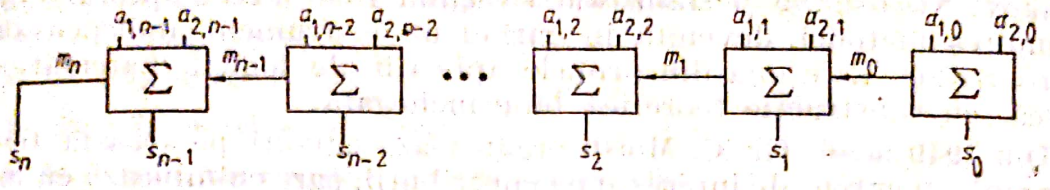


Fig. 5.44

numeroase și se referă nu numai la circuitele combinaționale. Este de notat însă aplicația lui la implementarea CLC-urilor. Pentru a ilustra această aplicație să considerăm o funcție logică de două variabile $f(x, y)$ a cărei tablă de valori este

x	y	$f(x, y)$
1	1	d_3
1	0	d_2
0	1	d_1
0	0	d_0

Reprezentarea Lagrange este

$$f(x, y) = d_3xy + d_2xy' + d_1x'y + d_0x'y'.$$

Coeficienții binari d_3, d_2, d_1, d_0 determină în mod unic pe f.

Cum există $2^4 = 16$ combinații de 0 și 1 pentru patru elemente, formula generală reprezintă 16 formule particulare, care corespund la tot atîtea circuite logice. Dacă formăm un circuit general ale cărui variabile sînt x, y, d_0, d_1, d_2, d_3, acesta poate fi organizat astfel ca în fig. 5.45.

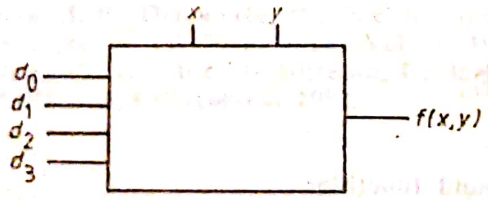


Fig. 5.45

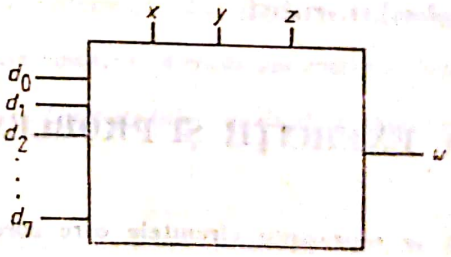


Fig. 5.46

Această schemă generală ne va permite implementarea oricăreia din cele 16 funcții logice, prin specificarea lui d_0, d_1, d_2, d_3 care îi sînt proprii. Astfel, pentru $d_3 = 0$, $d_2 = 1$, $d_1 = 1$, $d_0 = 0$, funcția devine

$$f(x, y) = x + y.$$

Avantajul de a implementa orice funcție cu ajutorul uneia și aceleiași scheme crește o dată cu numărul de variabile.

Principiul folosirii multiplexorului la implementarea (sub formă de circuit combinațional) oricărei funcții date este cel descris. Un multiplexor de opt intrări de date are schema din 5.46.

Scanned with OKEN Scanner

5.8 NOTĂ ISTORICĂ

Algebra circuitelor cu contacte a fost schițată în 1938 de C. E. Shannon (creatorul formulei de măsură a informației), care a arătat legătura ei cu calculul propozițiilor și cu algebra booleană. Printre pionieri se înscriu V. I. Șestakov, Nakashima și Hanzawa. Deceniul 1950 aduce aprofundarea și răspîndirea metodei, devenită în curînd un instrument indispensabil al electroniștilor. Este una din primele aplicații ale logicii matematice în tehnică, cu construcție teoretică bine închegată.

Din 1949 acad. Gr. C. Moisil organizează cursuri pe această temă și determină un curent de interes și de contribuții, care culminează cu cartea a în 1965, „Teoria algebrică a schemelor cu contacte și relee" tradusă în mai multe limbi. Tot în acest an la București se organizează un simpozion internațional sub egida IFAC (Federația Internațională de Control Automat), pregătit și găzduit de școala românească. Începînd cu 1956 Gr. C. Moisil aplică la teoria circuitelor logica polivalentă a lui Lukasiewicz, în timp ce V. I. Șestakov folosește sistemul lui Bochvar (v. cap. 7). N. Rescher îi citează ca deschizători ai acestui capitol de aplicație a logicii.

Aflate în generația a 4-a cu circuite, cu integrare largă (LSI) și integrare pe scară foarte largă (VLSI), calculatoarele reclamă o proiectare de mare complexitate a componentelor. Numai în deceniul 1971—1980 s-a trecut de la 1000 tranzistori pe pastile de siliciu cu 8000 operații/s la 68 000 tranzistori pe același suport cu 800 000 operații/s, lungimea operatorului fiind de 4 ori mai mare.

Pentru a proiecta circuitele dorite se recurge la programe la calculator și în ultimul timp la sisteme expert inteligente.

Deși fac parte integrantă din construcția calculatoarelor, circuitele combinaționale de care s-a vorbit în acest capitol nu reprezintă decît prima treaptă (de ordinul zero). Treptele următoare sînt constituite din circuite de memorare, circuite secvențiale, sincrone și asincrone, tratate în teoria automatelor finite, de importanță cardinală pentru arhitectura și funcționarea calculatorului.

5.9. EXERCIȚII ȘI PROBLEME

1. Să se reprezinte circuitele care corespund funcțiilor:

a) $xyz + xy'z' + x'yz' + x'y'z$,

b) $(x + y + z)(x' + y)(xz + x'y)(y' + z)$.

2. Ce reprezentare are circuitul a cărui funcție polinomială cu patru variabile este 0 în (0, 0, 1, 0), (0, 0, 1, 1,), (0, 1, 1, 0), (0, 1, 1, 1), (1, 0, 0, 0), (1, 0, 0, 1), (1, 1, 0, 0,) (1, 1, 0, 1), (1, 1, 1, 1) și 1 în rest.

3. Să se desemneze un sistem de închidere a unei conducte de petrol alimentate din trei surse prin trei supape de închidere S_1, S_2, S_3 astfel ca petrolul să curgă în situațiile: a) S_1 și S_3 închise sau deschise ambele, cu S_2 deschis; b) S_1 deschis, S_2 și S_3 închise.

4. Să se desemneze o mașină de vot pentru 3 votanți, cu contacte DA — NU, astfel ca să existe curent numai în cazul unei majorități de voturi și apoi simplificați circuitul. Dar pentru 4 și 5 votanți?

5. Să se reprezinte prin NAND pe ȘI, SAU și NON. Aceeași problemă prin NOR.

Scanned with OKEN Scanner

6. Presupunînd că se dispune numai de elemente NOR (cu 2 intrări) de NON, să se deseneze cele mai simple circuite pentru funcțiile $\bar{p}q \vee p\bar{q}$ și $pq \vee \bar{p}\bar{q}$.

Aceeași problemă cu elemente NAND.

7. Să se simplifice circuitul corespunzînd funcției

$$\bar{x}\bar{y}\bar{z} + \bar{x}yz + x\bar{y}z + \bar{x}y\bar{z} + \bar{x}\bar{y}z + xyz.$$

Cîte elemente s-au economisit?

8. Să se reprezinte și să se simplifice prin metoda Quine-McCluskey circuitele corespunzînd funcțiilor:

a) $\bar{x}\bar{y}z + \bar{x}yz + xyz + xy\bar{z}$,

b) $\bar{x}\bar{y}z + \bar{x}y\bar{z} + xy\bar{z} + \bar{x}yz + xyz + x\bar{y}z$.

9. Să se reprezinte circuitele ce corespund funcțiilor:

a) $x\bar{y}\bar{z} + x\bar{z}w + yzw + \bar{x}zw + \bar{x}\bar{y}\bar{z}$,

b) $\bar{x}yz\bar{w} + \bar{z}w + xy\bar{w} + \bar{x}yz\bar{w}$

și să se simplifice, recurgîndu-se la diagrame Karnaugh.

10. Să se simplifice circuitul corespunzînd funcției

$$xyz + x\bar{z} + x_1x_2\bar{w} + \bar{x}_1z + \bar{x}\bar{y}\bar{z}\bar{w},$$

folosind și comparînd procedeele Quine-McCluskey și al diagramelor Karnaugh. Cîte elemente sînt necesare pentru cele patru forme minimale?

BIBLIOGRAFIE

1. Blakeslee, G. *Digital Designs with Standard MSI and LSI.* John Wiley, 1979.
2. Goldschlager, L., Lister, A. *Computer Science.* Prentice-Hall, 1982.
3. Green, D. *Modern Logic Design.* Addison Wesley Publ. Co., 1985.
4. Marley, G., Earle, J. *The Logic Design of Transistor Digital Computers.* Englewood Cliffs, Prentice-Hall, 1963.
5. Moisil, Gr. C. *Teoria algebrică a schemelor cu contacte și relee.* București, Editura Tehnică, 1970.
6. Perrin, J. P., Denouette, M., Daclin. *Méthodes modernes d'étude des systèmes logiques.* Ecole supérieure de l'Aeronautique, Vol. I, 1966.
7. Ștefan, Gh., Drăghici, I., Mureșan, T., Barbu, E. *Circuite integrate digitale.* București, Editura Didactică și Pedagogică, 1983.

Scanned with OKEN Scanner

6.
INTRODUCERE ÎN LOGICA INTUIȚIONISTĂ ȘI LOGICA POZITIVĂ

6.1. CONCEPTE FUNDAMENTALE

Logica intuiționistă este o logică neclasică care nu recunoaște principiul „terțului exclus" : $A \vee \sim A$ nu este o teză și prin aceasta bivalența (o propoziție este sau adevărată sau falsă, a treia posibilitate fiind exclusă), care caracterizează logicile clasice, nu funcționează. Ea s-a născut din curentul intuiționist în matematică, care recunoaște drept „aserțiuni" (teze, axiome sau teoreme) numai acele enunțuri, pentru care există o metodă de construcție a obiectului matematic la care se referă enunțul. În sensul intuționist „a aserta" este echivalent cu „a construi". Notația $\vdash A$ se citește „A este asertată" sau „ am efectuat o construcție A" sau „există o metodă de a construi pe A".

Conectorii unei logici propoziționale intuiționiste sînt tot conjuncția $\wedge$, disjuncția $\vee$, negația $\neg$ și implicația $\rightarrow$, definiți astfel :

1) Putem aserta $A \wedge B$ dacă și numai dacă atît A cît și B pot fi asertate (echivalent : am construit atît pe A cît și pe B).

2) Putem aserta $A \vee B$ dacă și numai dacă cel puțin una din propozițiile A și B pot fi asertate (echivalent : una din ele are construcție).

3) Putem aserta $\neg A$, dacă și numai dacă avem o construcție care, presupunînd pe A, conduce la o contradicție.

4) Putem aserta $A \rightarrow B$, dacă și numai dacă sîntem în posesia unei construcții care, asociată cu construcția lui A, conduce la o construcție a lui B.

Prin aceste definiții, toți conectorii sînt operații primitive care nu se pot defini unii cu ajutorul altora și nici prin recurgerea la tablele de adevăr (nu sînt funcționale de adevăr).

Contrastul cel mai semnificativ cu calculul propozițional clasic îl prezintă negația intuiționistă $\neg$. Negația clasică $\sim$ se definește în funcție de valorile de adevăr : $\sim p$ este fals numai cînd p este adevărat. În logica intuiționistă $\neg A$ înseamnă că A este falsă. Intuiționistic, negația este identică cu falsul. În timp ce $\neg A$ se citește : „este imposibil ca A să fie asertată", „nu se poate ca A să fie asertată" sau „pornind de la A ajungem la o contradicție", negația $\sim A$ înseamnă „nu știm dacă A" sau „nu sîntem îndreptățiți să afirmăm A". Din definiția negației 3) rezultă că putem considera că $\neg A$ implică falsul (contradicția) :

$$\neg A = A \rightarrow 0.$$

Această relație împreună cu faptul că $A \vee \neg A$ nu este o aserțiune (în sensul clasic o tautologie : $A \vee \neg A \neq 1$) ne dezvăluie natura algebrică

Scanned with OKEN Scanner

a negației în sensul intuiționist : ea nu este un complement, ci un *pseudocomplement.*

Fundamental pentru logica intuiționistă este respingerea valabilității universale a lui $A \vee \neg A$ ca aserțiune. Dacă această formulă ar fi o aserțiune, ar însemna sau să avem o singură metodă pentru a ne pronunța dacă A sau $\neg A$ sînt asertabile (conform definiției disjuncției), dar construcția lui A și construcția unei contradicții din presupunerea lui A sînt două lucruri diferite (conform definiției negației). *Tertium non datur* este respins de intuiționiști ca tautologie.

Definițiile 3) și 4) sînt suportul celei de-a doua importante poziții a intuiționiștilor, legate de dubla negație. În logica clasică $\sim\sim A \equiv A$, dar formula nu este valabilă în logica intuiționistă, care acceptă

$$A \to \neg\neg A,$$

dar respinge

$$\neg\neg A \to A.$$

Exemple. a) Fie $A \underset{\sim}{\triangle}$ „Am construit numerele întregi x, y, z, m astfel ca pentru $m > 2$, $x^m + y^m = z^m$“. Nu putem aserta A, deoarece teorema lui Fermat nu este demonstrată. Dar din presupunerea că construcția din A a fost efectuată nu putem obține o contradicție, deci nu putem aserta $\neg A$. Astfel nici $A \vee \neg A$ nu este asertabilă.

b) Fie $\pi = 3{,}14159\ldots$ și $\rho = 0{,}33333\ldots$. Cînd în dezvoltarea lui π întîlnim secvența 0123456789 (nu se știe dacă aceasta se întîmplă), oprim dezvoltarea lui ρ, punînd zerouri după secvența de 3. Fie acum $A \underset{\sim}{\triangle}$ „ρ este un număr rațional“. Din punct de vedere clasic A este adevărată. Din punct de vedere intuiționistic, A se enunță : „putem să construim numerele întregi m și n astfel ca $\rho = m/n$“. Presupunînd $\neg A$, obținem o contradicție : ρ nu poate fi de forma $0{,}333\ldots 300\ldots$ și $\rho = 1/3$. Intuiționistic, avem $\neg\neg A$. Dar A nu poate fi asertat intuiționist și deci nu este valabilă implicația $\neg\neg A \to A$ (Brouwer).

6.2. LOGICA INTUIȚIONISTĂ ȘI ALGEBRELE PSEUDOBOOLENE

Introducem un calcul propozițional intuiționist pe un limbaj de ordinul 0 : $S_i = \{L_0, C_i\}$ în mod analog cu un calcul propozițional clasic. Luăm ca punct de plecare axiomele $T1 - T12$ ale calculului propozițional (v. 3.11), suprimînd pe ultima

$$T12 \quad A \vee \sim A.$$

Rescriem formulele $T1 - T11$ în notațiile adoptate : $\wedge$, $\vee$, $\neg$, $\to$ și le considerăm axiome ale calculului intuiționist :

$$\begin{aligned}
&T1 \quad (A \to B) \to ((B \to C) \to (A \to C)),\\
&T2 \quad A \to A \vee B,\\
&T3 \quad B \to A \vee B,\\
&T4 \quad (A \to C) \to ((B \to C) \to ((A \vee B) \to C)),\\
&T5 \quad A \wedge B \to A,\\
&T6 \quad A \wedge B \to B,\\
&T7 \quad (C \to A) \to ((C \to B) \to (C \to (A \wedge B))),\\
&T8 \quad (A \to (B \to C)) \to ((A \wedge B) \to C),
\end{aligned}$$

Scanned with OKEN Scanner

T9 $((A \wedge B) \to C) \to (A \to (B \to C))$,

T10 $(A \wedge \neg A) \to B$,

T11 $(A \to (A \wedge \neg A)) \to \neg A$.

Operația de consecință $C_i(S)$ se introduce la fel ca în cazul calculului propozițional clasic (v. 2.10).

O secvență finită de formule este o *demonstrație intuiționistă* sau o *deducție (derivație) intuiționistă* la fel ca în definițiile din 2.1, cu o singură specificație că prin axiome se înțeleg axiomele sistemului intuiționist T1—T11.

O *formulă* A este *intuiționist deductibilă*, dacă se află în $C_i(0)$.

Orice formulă deductibilă intuiționistic este clasic deductibilă.

O *teorie propozițională intuiționistă* este de forma $\mathscr{T} = \{\mathscr{L}_0, C_i, \mathscr{A}\}$, unde $\mathscr{A}$ este mulțimea sa de axiome. Formulele $C_i(\mathscr{A})$ sînt teoremele teoriei $\mathscr{T}$.

În mod analog cu procedura de la 3.5 și 3.11 considerăm algebra formulelor $\{F, \vee, \wedge, \to, \neg\}$ corespunzătoare teoriei intuiționiste. Relația $\approx$ definită prin :

$A \approx B$ dacă și numai dacă $A \to B$ și $B \to A$ sînt teoreme în $\mathscr{T}$, este o relație de congruență în algebra $\{F, \vee, \wedge, \to, \neg\}$. Algebra $F/\approx$ este algebra teoriei și o notăm cu $\mathscr{A}(\mathscr{T})$.

Algebra pseudobooleană joacă același rol pentru o teorie propozițională intuiționistă ca și algebra booleană pentru o teorie propozițională clasică.

Se demonstrează (v. 3.11) că

Algebra $\mathscr{A}(\mathscr{T})$ a unei teorii propoziționale intuiționiste este o algebră pseudobooleană. Oricare ar fi formulele A, B, avem :

$$|A| \vee |B| = |A \vee B|$$
$$|A| \wedge |B| = |A \wedge B|$$
$$|A| \to |B| = |A \to B|$$
$$\neg |A| = |\neg A|$$

Relația $|A| \leqslant |B|$ are loc dacă și numai dacă $A \to B$ este o teoremă în $\mathscr{T}$. O formulă A este o teoremă în $\mathscr{T}$ dacă și numai dacă elementul $|A|$ este elementul 1 în $\mathscr{A}(\mathscr{T})$. Teoria este consistentă dacă și numai dacă algebra pseudobooleană $\mathscr{A}(\mathscr{T})$ este nedegenerată.

Rămîn valabile definițiile stabilite în logică clasică pentru evaluare și model. O formulă A este validă în algebra pseudobooleană B, dacă fiecare evaluare $v \in B^P$ este un model pentru A :

$$A_B^v = 1.$$

(Am notat aplicația $v : P \to B$, unde P este mulțimea variabilelor propoziționale, iar B algebra pseudobooleană și

$$A_B : B^P \to B$$

aplicația determinată de o formulă formată din variabilele propoziționale P.)

O formulă este *intuiționist validă* sau este o *tautologie intuiționistă*, dacă este validă în orice algebră pseudobooleană.

Scanned with OKEN Scanner

Pentru a stabili teorema de echivalență între tautologii și teoreme în logica intuiționistă, procedăm ca și în cazul clasic.

a) Demonstrăm mai întîi că orice teoremă (formulă deductibilă) este tautologie intuiționistă. Este suficient să arătăm că axiomele T1—T11 sînt tautologii intuiționiste și că modus ponens conservă această proprietate. Cu titlu de exemplu, pentru T1, notăm

$$a = A_B^v, \qquad b = B_B^v, \; c = C_B^v,$$

unde indicele reprezintă logica pseudobooleană. Apoi

$$[(A \to B) \to ((B \to C) \to (A \to C))]_B^v = (a \to b) \to ((b \to c) \to (a \to c)) = 1.$$

b) Reciproca se demonstrează analog ca în 3.11 : orice tautologie intuiționistă (validă) este deductibilă în calculul propozițional intuiționist. În acest fel avem o teoremă de completitudine pentru calculul propozițional intuiționist.

Următoarele formule sînt tautologii intuiționiste :

T1′ $A \to A$
T2′ $A \to (B \to A)$
T3′ $A \to (B \to (A \wedge B))$
T4′ $(A \to (B \to C)) \to (B \to (A \to C))$
T5′ $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
T6′ $A \to \neg\neg A$
T7′ $\neg(A \wedge \neg A)$
T8′ $(\neg A \vee B) \to (A \to B)$
T9′ $\neg(A \vee B) \to (\neg A \wedge \neg B)$
T10′ $\neg A \wedge \neg B \to \neg(A \vee B)$
T11′ $\neg A \vee \neg B \to \neg(A \wedge B)$
T12′ $(A \to B) \to (\neg B \to \neg A)$
T13′ $(A \to \neg B) \to (B \to \neg A)$
T14′ $\neg\neg\neg A \to \neg A$
T15′ $\neg A \to \neg\neg\neg A$
T16′ $\neg\neg(A \to B) \to (A \to \neg\neg B)$
T17′ $(C \to A) \to ((C \to (A \to B)) \to (C \to B))$.

Demonstrația este la fel cu cea indicată la demonstrarea caracterului tautologic al axiomelor T1—T11, a). Întrucît tautologiile sînt și teoreme derivabile, pot fi demonstrate cu ajutorul lui MP.

Este important să recunoaștem care formule nu sînt tautologii intuiționiste, deși ele sînt tautologii în sens clasic.

Următoarele formule nu sînt tautologii intuiționiste :

F1 $p \vee \neg p$
F2 $\neg\neg p \to p$
F3 $(p \to q) \to (\neg p \vee q)$
F4 $\neg(p \wedge q) \to (\neg p \vee \neg q)$
F5 $(\neg p \to q) \to (\neg q \to p)$
F6 $(\neg p \to \neg q) \to (q \to p)$
F7 $((p \to q) \to p) \to p$

Pentru demonstrație este suficient să găsim o algebră pseudobooleană B și o evaluare în B, astfel ca formulele Fi să aibă o evaluare $F_B^v \neq 1$. Luăm drept algebra B algebra mulțimilor deschise de numere reale. Pentru F1 : $v(p) = S$, unde S este $(0, \infty)$. Se vede că $F_1^v \neq 1$.

6.3. AXIOMATIZĂRI

Alegerea axiomelor T1—T11 (v. 6.2) este determinată de intenția de a scoate în evidență structura algebrică corespunzătoare sistemului logic. Alte mașini de axiome pentru logica intuiționistă cunoscute în literatură sînt :

a) **Sistemul lui Heyting**

H1 $a \to (a \wedge a)$
H2 $(a \wedge b) \to (b \wedge a)$
H3 $(a \to b) \to ((a \wedge c) \to (b \wedge c))$
H4 $((a \to b) \wedge (b \to c)) \to (a \to c)$
H5 $b \to (a \to b)$
H6 $(a \wedge (a \to b)) \to b$
H7 $a \to (a \vee b)$
H8 $(a \vee b) \to (b \vee a)$
H9 $(a \to c) \wedge (b \to c) \to ((a \vee b) \to c)$
H10 $\neg a \to (a \to b)$
H11 $((a \to b) \wedge (a \to \neg b)) \to \neg a$

Metoda de care s-a servit Heyting pentru independența axiomelor este cea a lui Bernays (v. 2.4). De exemplu, matricele cu care stabilește independența axiomei H1 sînt :

$\to$	0	1	2
0	0	0	0
1	2	0	2
2	2	0	0

$\wedge$	0	1	2
0	0	1	1
1	1	1	1
2	1	1	1

$\vee$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

$\neg$	0	1	2
	1	0	1

Se stabilesc prin MP următoarele teoreme :

T1'' $a \to \neg\neg a$
T2'' $(a \to b) \to (\neg a \to \neg b)$
T3'' $(a \to b) \to (\neg\neg a \to \neg\neg b)$
T4'' $\neg a \to \neg\neg\neg a$
T5'' $\neg\neg\neg a \to \neg a$
T6'' $\neg(a \vee b) \to (\neg a \wedge \neg b)$
T7'' $\neg a \wedge \neg b \to \neg(a \vee b)$
T8'' $\neg a \vee \neg b \to \neg(a \wedge b)$
T9'' $\neg\neg(a \wedge \neg a)$
T10'' $\neg\neg(a \wedge b) \to (\neg\neg a \wedge \neg\neg b)$

Scanned with OKEN Scanner

T11'' $(\neg\neg a \wedge \neg\neg b) \to \neg\neg(a \wedge b)$
T12'' $(\neg\neg a \vee \neg\neg b) \to \neg\neg(a \vee b)$
T13'' $(\neg\neg a \wedge \neg\neg(a \to b)) \to \neg\neg a$
T14'' $(\neg\neg(a \to b) \wedge \neg\neg(b \to c) \to \neg\neg(a \to c)$

b) **Sistemul lui Johansson.** Eliminînd din sistemul lui Heyting axioma H10 şi menţinînd H1, H9, H11 ca axiome, se obţine *sistem intuiţionist* numit *minimal.*

În calculul clasic H10 este o tautologie. Dar în logica intuiţionistă justificarea sa este dificilă : din ipoteza că am putea construi pe a obţinem o contradicţie, fapt care asociat cu construcţia inexistentă a lui a ar putea demonstra pe b.

Renunţînd la H10, sistemul minimal nu mai conţine drept teoreme următoarele formule care sînt teoreme în sistemul lui Heyting :

F1 $(a \wedge \neg a) \to b$
F2 $(a \wedge \neg a) \vee b \to b$
F3 $(a \vee b) \wedge \neg a \to b$
F4 $(b \vee \neg b) \to (\neg\neg b \to b)$
F5 $(\neg a \vee b) \to (a \to b)$
F6 $(a \vee b) \to (\neg a \to b)$
F7 $((a \to b) \vee \neg c) \to ((a \wedge c) \to b)$
F8 $\neg\neg(\neg\neg a \to a)$

c) **Axiomatizarea lui Lukasiewicz.** Lukasiewicz a reaxiomatizat în felul următor axiomele lui Heyting :

L1 $(A \wedge B) \to A$
L2 $(A \wedge B) \to B$
L3 $A \to (A \vee B)$
L4 $B \to (A \vee B)$
L5 $A \to (B \to A)$
L6 $A \to (\neg A \to B)$
L7 $A \to (B \to (A \wedge B))$
L8 $(A \to \neg B) \to (B \to \neg A)$
L9 $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
L10 $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

d) **Sistemul lui Kleene.** Kleene scrie axiomele pentru calculul propoziţional clasic, în care înlocuieşte doar axioma dublei negaţii prin K10 (H10) :

K1 $p \to (q \to p)$
K2 $(p \to q) \to ((p \to (q \to r)) \to (p \to r))$
K3 $p \to (q \to (p \wedge q))$
K4 $(p \wedge q) \to p$
K5 $(p \wedge q) \to q$
K6 $p \to (p \vee q)$
K7 $q \to (p \vee q)$
K8 $(p \to r) \to ((q \to r) \to ((p \vee q) \to r))$

Scanned with OKEN Scanner

K9 $(p \to q) \to ((p \to \neg q) \to \neg p)$

K10 $\neg p \to (p \to q)$

Dacă în loc de K10 repunem la loc $\neg\neg p \to p$, obținem calculul propozițional clasic.

e) **Sistemul lui Troelstra.** În acest sistem mai multe axiome sînt enunțate sub formă de reguli :

1) $A \to A$

2) $A \wedge B \to A$; $B \wedge A \to A$

3) $A \to A \vee B$; $B \to A \vee B$

4) $0 \to A$

5) $\dfrac{A,\ A \to B}{B}$

6) $\dfrac{A \to B,\ B \to C}{A \to C}$

7) $\dfrac{A \to C,\ B \to C}{(A \vee B) \to C}$

8) $\dfrac{C \to A,\ C \to B}{C \to (A \wedge B)}$

9) $\dfrac{(A \wedge B) \to C}{A \to (B \to C)}$

10) $\dfrac{A \to (B \to C)}{(A \wedge B) \to C}$

Prin adăugarea la acest sistem a legii terțului exclus $A \vee \neg A$ se obține calculul propozițional clasic. Formula $\neg A$ este prin definiție $A \to 0$.

f) **Sistemul lui Kolmogorov.** Kolmogorov a considerat următoarele axiome :

K1′ $p \to (q \to p)$

K2′ $(p \to (p \to q)) \to (p \to q)$

K3′ $(p \to (q \to r)) \to (q \to (p \to r))$

K4′ $(q \to r) \to ((p \to q) \to (p \to r))$

K5′ $(p \to q) \to ((p \to \neg q) \to \neg p)$

Primele patru axiome coincid cu axiomele lui Hilbert pentru implicația în calculul propozițional clasic. Axioma K5′ este principiul contradicției, pentru că poate fi citit astfel : dacă din p urmează atît q, cît și $\neg q$, atunci p este fals". Axioma K5′ înlocuiește axiomele negației din calculul clasic :

$$p \to (\sim p \to q),\ (p \to q) \to ((\sim p \to q) \to q).$$

Prezintă interes interpretarea dată de Kolmogorov sistemului intuiționist al lui Heyting drept un „calcul al problemelor".

g) **Sistemul lui Glivenko.** Anterior lui Heyting, Glivenko a elaborat un tabel de axiome :

G1 $p \to p$

G2 $(p \to q) \to ((q \to r) \to (p \to r))$

G3 $(p \wedge q) \to p$

G4 $(p \wedge q) \to q$

G5 $(r \to p) \to ((r \to q) \to (r \to (p \wedge q)))$

G6 $p \to (p \vee q)$

G7 $q \to (p \vee q)$

G8 $(p \to r) \to ((q \to r) \to ((p \vee q) \to r))$

G9 $(p \to q) \to ((p \to \neg q) \to \neg p)$

Scanned with OKEN Scanner

G10 $(p \to \neg q) \to (q \to \neg p)$
G11 $(p \to q) \to (\neg q \to \neg p)$
G12 $(p \to (q \to r)) \to (q \to (p \to r))$
G13 $(p \to (p \to r)) \to (p \to r)$
G14 $p \to (q \to p)$
G15 $\neg q \to (q \to p)$

Prin simpla adăugare a axiomei $p \vee \neg p$ rezultă calculul clasic.

6.4. LOGICĂ POZITIVĂ ȘI LATICE RELATIV PSEUDOCOMPLEMENTATE

Luînd ca punct de plecare calculul propozițional clasic, calculul propozițional intuționist este obținut prin eliminarea unor axiome. Limbajul de ordinul zero a rămas neschimbat, deși negației (notată altfel) i s-au impus reguli diferite de către noile tabele de axiome.

Vom continua procedeul, eliminînd noi axiome din lista inițială și aducînd de această dată modificări și alfabetului. Vom adopta astfel un alfabet care nu conține decît conectorii $\vee$, $\wedge$, $\to$. Absența negației face ca sistemul logic construit pe un asemenea limbaj să poarte denumirea de *logică pozitivă*.

Din lista de axiome T1—T11 (v. 6.2) eliminăm ultimele două, care se referă ambele la negație. Considerăm axiomele T1—T9 ca bază a logicii pozitive.

În mod analog, la calculul propozițional clasic și cu calculul propozițional intuționist, definim calculul propozițional pozitiv

$$S_p = \{\mathscr{L}_0, C_p\}$$

și teoriile pozitive de ordinul zero

$$\mathscr{T} = \{\mathscr{L}_0, C_p, \mathscr{A}_p\}.$$

Formulele din mulțimea consecințelor logice $C_p(\mathscr{A}_p)$ vor fi teoreme și analog cu noțiunea introdusă în 6.2 vom avea formule pozitiv deductibile, care se află în $C_p(0)$.

Algebra asociată a formulelor va fi

$$\{F, \cup, \cap, \to\}$$

și va avea trei operații. Toate rezultatele anterioare, în afara celor care conțin pe $\neg$, rămîn valabile în sistemul pozitiv.

Relația $\approx$ definită în F:

$A \approx B$, dacă și numai dacă $A \to B$ și $B \to A$ sînt teoreme,

este o congruență în algebra formulelor. Algebra teoriei $F/\approx$ se notează ca și pînă acum cu $\mathscr{A}(\mathscr{T})$. Avem următorul rezultat (v. 3.11):

Algebra $\mathscr{A}(\mathscr{T})$ a unei teorii propoziționale pozitive este o latice relativ pseudocomplementată. Oricare ar fi formulele A, B, avem:

$$|A| \vee |B| = |A \vee B|, \quad |A| \wedge |B| = |A \wedge B|, \quad A \to |B| = |A \to B|$$

Scanned with OKEN Scanner

Relația $|A| \leqslant |B|$ are loc dacă și numai dacă $A \to B$ este o teoremă în $\mathscr{T}$. Elementul $|A|$ este 1 în $\mathscr{A}(\mathscr{T})$ dacă și numai dacă A este o teoremă în $\mathscr{T}$.

Laticea relativ pseudocomplementată $\mathscr{F}/\approx$ servește la definirea unei evaluări în această algebră, a unui model și a validității. O *formulă* este *pozitiv validă* sau o *tautologie propozițională pozitivă* dacă este validă (orice evaluare este model) în laticea relativ pseudocomplementată.

O teoremă de completitudine, analoagă cu cea stabilită pentru calculul intuiționist afirmă că într-un sistem propozițional pozitiv, pentru orice formulă deductibilitatea în sistem (caracterul de teoremă) este echivalent cu proprietatea de tautologie propozițională pozitivă.

Examinarea calculului pozitiv poate fi redusă la examinarea acelei părți a calculului intuiționist care cuprinde formule fără conectorul $\neg$.

Formulele T1′—T5′ sînt tautologii pozitive, în timp ce F7 nu este (v. 6.2).

Axiomele T1—T9 au fost selectate pentru că scot mai bine în evidență structura algebrică asociată sistemului logic. Se recurge des la *logica pozitivă Hilbert-Bernays*, caracterizată de următorul sistem de axiome:

HB1	$A \to (B \to A)$
HB2	$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
HB3	$(A \wedge B) \to A$
HB4	$(A \wedge B) \to B$
HB5	$(A \to B) \to ((A \to C) \to (A \to (B \wedge C)))$
HB6	$A \to (A \vee B)$
HB7	$B \to (A \vee B)$
HB8	$(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

și MP.

Logicile pozitive ale lui Moisil. Pornind de la sistemul Hilbert-Bernays, Moisil a arătat prin ce adăugări de axiome din acesta se obțin sistemele intuiționiste ale lui Kolmogorov, Johanssen, Glivenko și Heyting sau sistemul clasic.

Moisil definește o logică strict pozitivă mai generală decît cea a lui Hilbert-Bernays cu doi conectori, conjuncție și disjuncție, fără distributivitate. În acest fel logica strict pozitivă stă la originea a două extensii posibile: a) prin adăugarea distributivității ea devine logică strict pozitivă distributivă, care conduce prin adăugarea principiului terțului exclus la logica clasică și b) în altă direcție ea devine logică strict simetrică, care prin adăugarea principiului terțului exclus conduce la *logica cuantică*.

Sistemul lui Griss. Ocupîndu-se numai de propoziții matematice, Griss a elaborat un sistem fără negație bazat pe axiomele

G1′	$p \to (p \wedge p)$
G2′	$(p \wedge q) \to (q \wedge p)$
G3′	$(p \to q) \to ((p \wedge r) \to (q \wedge r))$
G4′	$((p \to q) \wedge (q \to r)) \to (p \to r)$
G5′	$(p \wedge q) \to p$

Scanned with OKEN Scanner

precum și regulile $\frac{P, Q}{P \wedge Q}$, $\frac{P, P \to Q}{Q}$, $\frac{R, P \to Q}{P \to Q \wedge R}$

6.5. NOTĂ ISTORICĂ

Logica intuiționistă își are originea în curentul intuiționist, opus formalismului, prin tezele sale asupra fundamentelor matematicii.

Primele poziții ale intuiționismului în matematică au fost susținute de L. E. J. Brouwer în 1907, dar sub formă de școală coerentă s-a constituit după 1920. Una din tezele intuiționiste care face distincția între matematică și limbaj fusese enunțată încă de Poincaré : „există o realitate mai subtilă care determină entitățile matematice și care este altceva decît logica" (1909). Matematica se ocupă de construcțiile matematice intuitive, în timp ce logica matematică se ocupă de limbajul în care sînt exprimate construcțiile matematice. Raportul care se constituie între ele nu este acela al logicii aplicate la fundamentarea sau construcția formală a matematicii, ci, invers, matematicile construite intuitiv sînt fundamentele logicii. Discuția s-a purtat în special în jurul teoriei mulțimilor infinite, intuiționiștii manifestîndu-și rezervele față de teorii axiomatice moderne ale mulțimilor. Centrală a fost și dezbaterea asupra noțiunii de *existență matematică*. Existența matematică implică necontradicție logică, dar necontradicția logică nu implică întotdeauna existența matematică efectivă, susține școala intuiționistă.

Poziția netă a intuiționiștilor împotriva recunoașterii valabilității universale a principiilor *tertium non datur* și *negatio duplex* a inspirat cercetările logicienilor după 1920 în direcția logicii neclasice, la care au participat curînd și matematicienii intuiționiști. De fapt, intuiționiștii nu se ridică împotriva teoriilor logice formale, ci împotriva rolului acestora. Intuiționiștii s-au preocupat de eliminarea paradoxurilor care au acompaniat elaborarea teoriilor formal logice și matematice.

Sistemele descrise în acest capitol au următoarea cronologie : Kolmogorov a elaborat sistemul său de axiome în 1925 ; Glivenko în 1928 ; Heyting în 1930 cu variante noi în 1966 ; Johansson în 1936 : axiomatizarea lui Lukasiewicz datează din 1953, sistemul lui Kleene este descris în cartea sa din 1967, iar sistemul lui Troelstra în cartea sa din 1970.

În ceea ce privește logica pozitivă, sistemul Hilbert-Bernays se găsește în cartea „*Grundlagen der Mathematik*" (1934 și în ediția mai recentă din 1970). Griss publică sistemul său în 1951, iar Moisil considerațiile citate în 1958 și 1965. Încadrarea logicii intuiționiste într-o logică modală generală a fost făcută de Moisil în 1940.

Interesul actual pentru logicile intuiționiste este determinat, pe de o parte, de curentul constructivist în matematică, curent în plină ascensiune (Bishop, Rice și alții) și, pe de altă parte, de efortul inteligenței artificiale de a modela raționamentul logic uman caracterizat de polivalență.

6.6. EXERCIȚII ȘI PROBLEME

1. Să se arate că toate axiomele logice T1 — T11 (v. 6.2) sînt tautologii intuiționiste.
2. Să se demonstreze că formulele T1′ — T17′ (v. 6.2) sînt tautologii intuiționiste.

Scanned with OKEN Scanner

3. Să se demonstreze că F2 — F7 (v. 6.2) nu sînt tautologii intuiționiste, folosind următoarele evaluări:

F2 $v(p) = S \cup T$
F3 $v(p) = S, \quad v(q) = S$
F4 $v(p) = S, \quad v(q) = T$
F5 $v(p) = S \cup T, \quad v(q) = 0$
F6 $v(p) = S \cup T, \quad v(q) = \mathbb{R}$
F7 $v(p) = S \cup T, \quad v(q) = 0$

unde $\mathbb{R}$ este mulțimea tuturor numerelor reale, S intervalul infinit deschis $(0, \infty)$, T intervalul infinit deschis $(-\infty, 0)$.

4. Să se arate că mulțimea formulelor intuiționist valide este închisă față de MP.

5. Să se arate că $\neg\neg a \to a$ este independentă de axiomele sistemului lui Heyting, folosind matricele:

$\to$	0	1	2
0	0	0	0
1	1	0	1
2	2	0	0

$\wedge$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

$\vee$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

$\neg$	0	1	2
	1	0	1

6. Să se compare lista teoremelor deduse din T1—T11 (v. 6.2) și lista teoremelor deduse din axiomele lui Heyting (v. 6.3). Sînt cele două liste unificabile?

7. Să se pună semnul implicației ($\to$ sau $\leftarrow$) între următoarele perechi de expresii:

a) $\neg a \vee b \qquad a \to b$
b) $\neg a \to b \qquad a \vee b$
c) $a \vee b \qquad \neg(\neg a \wedge \neg b)$
d) $\neg(\neg a \vee \neg b) \qquad a \wedge b$
e) $(a \to b) \to b \qquad a \vee b$

astfel ca să se obțină o tautologie intuiționistă.

8. Pot fi teoremele T13'' și T14'' (v. 6.3) (demonstrate de Moisil în sistemul lui Heyting) deduse în sistemul T1 — T11 (v. 6.2) sau în sistemul intuiționist Lukasiewicz (v. 6.3)?

9. Sînt formulele F1 — F7 teoreme în sistemele Heyting sau Lukasiewicz?

10. Să se studieze echivalența sistemelor definite de axiomele T1 — T9 și de axiomele HB1 — HB8 (v. 6.4).

BIBLIOGRAFIE

1. Dragalin, A. G. *Matematiceskii intuiționizm.* Moscova, 1979.
2. Gödel, K. *Eine Interpretation des intuitionistischen Aussagenkalkül.* În: Ergebnisse eines mathematischen Kolloquium. Heft 4, 1933.
3. Griss, G.F.C. *La mathémathique intuitioniste sans négation.* Nieuw Archief voor Wiskunde, III, 1955.
4. Heyting, A. *Intuitionism. An Introduction.* Amsterdam, Studies, in Logic and Foundations of Mathematics, 1956.
5. Hilbert, D., Bernays, P. *Grundlagen der Mathematik.* Berlin, Springer Verlag, 1970.
6. Joja, A. *About Tertium Non Datur.* București, Acta logica, Vol. I, 1958.
7. Kolmogorov, A. N. *On the Principle of Excluded Middle.* În: From Frege to Gödel, Harward Univ. Press, 1967.
8. Lorenzen, P. *Formal Logic.* Reidel, 1965.
9. Moisil, Gr. C. *Sur la logique positive.* București, Acta logica, Vol. I, 1958.
10. Moisil, Gr. C. *Sur la logique strictement positive.* În: Analele Universității Al. I. Cuza, XI, 15, 1965.

Scanned with OKEN Scanner

11. Mostowski, A. *Various Interpretations of the Intuitionistic Logic.* În : Acta Philosophica Fenica, XVII, 1965.
12. Myhill, J. *The Formalization of Intuitionism.* În : La philosophie contemporaine. Firenze, 1968.
13. Onicescu, O. *Principes de logique et de philosophie mathématique.* București, Editura Academiei R.S.R., 1971.
14. Rasiowa, H. *An Algebraic Approach to Non-Classical Logics.* Warszawa, PWN, 1974.
15. Surdu, A. *Elemente de logică intuiționistă.* București, Editura Academiei R.S.R., 1976.

Scanned with OKEN Scanner

7.
ELEMENTE DE LOGICĂ POLIVALENTĂ

7.1. CONSTRUCȚIA SEMANTICĂ

Punctul de plecare în construirea unei logici polivalente presupune părăsirea ipotezei că propozițiile p, q, r, ... pot lua numai două valori de adevăr: adevărat (1) și fals (0). Noua ipoteză, în cazul logicii trivalente, va admite că p, q, r, ... pot lua trei valori de adevăr: adevărat(2), nesigur (incert, posibil) (1) și fals (0). O propoziție p care va avea valoarea 1 se va citi: „poate că p“. Evaluarea în această logică va fi o aplicație a mulțimii propozițiilor pe mulțimea $\{0, 1, 2\}$.

Este normal ca logica trivalentă să fie privită drept o generalizare a logicii bivalente. De aceea adoptăm aceleași definiții pentru conectorii $\neg$, $\wedge$, $\vee$. Tablele de adevăr prin care au fost introduși acești conectori în logica bivalentă confirmă următoarele definiții:

$$v(\neg p) = 1 - v(p),$$

$$v(p \wedge q) = \min(v(p), v(q)),$$

$$v(p \vee q) = \max(v(p), v(q)).$$

Adoptăm aceste definiții pentru noua funcție de evaluare, cu diferența că locul lui 1 fiind luat de 2, $v(\neg p) = 2 - v(p)$. Rezultă următoarele table de adevăr

p	$\neg p$
0	2
1	1
2	0

p \ q	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

($p \wedge q$)

p \ q	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

($p \vee q$)

Observăm că renunțînd la valoarea intermediară, tăind peste tot linia și coloana corespunzătoare lui 1 și înlocuind pe 2 cu 1, obținem tablele de adevăr pentru $\neg$, $\wedge$ și $\vee$ din logica bivalentă.

Tablele de adevăr pentru logica trivalentă vor fi matrice cu $3^2 = 9$ elemente sau table cu $3^2 = 9$ linii.

Deși definiția evaluării s-a schimbat, restul noțiunilor definite în logica bivalentă rămîne același: regulile de formare a formulelor, funcții

Scanned with OKEN Scanner

de adevăr (tablele), tautologii (formule care iau întotdeauna valoarea 2) și contradicții, echivalențe. Principalul element de lucru rămîne tabla de adevăr. Cu ajutorul ei constatăm că formula De Morgan se menține:

p	q	$\neg p$	$\neg q$	$\neg p \vee \neg q$	$\neg(\neg p \vee \neg q)$	$p \wedge q$
0	0	2	2	2	0	0
0	1	2	1	2	0	0
0	2	2	0	2	0	0
1	0	1	2	2	0	0
1	1	1	1	1	1	1
1	2	1	0	1	1	1
2	0	0	2	2	0	0
2	1	0	1	1	1	1
2	2	0	0	0	2	2

Putem scrie $p \wedge q = \neg(\neg p \vee \neg q)$.

În schimb, așa cum era de așteptat, legea contradicției și legea terțului exclus din logica bivalentă nu mai sînt valabile:

p	$\neg p$	$p \wedge \neg p$	$p \vee \neg p$
0	2	0	2
1	1	1	1
2	0	0	2

Se observă că $p \wedge \neg p$ nu este întotdeauna falsă, deoarece pentru $p = 1$ nu ia valoarea 0, iar $p \vee \neg p$ nu este întotdeauna adevărată, deoarece pentru $p = 1$ nu ia valoarea 2.

În logica trivalentă nu este valabil principiul terțului exclus, ci principiul cvartului exclus

$$p \vee (\neg p) \vee (\neg(\neg p)) = 2,$$

iar principiul contradicției se enunță

$$p \wedge (\neg p) \wedge (\neg(\neg p)) = 0.$$

Verificăm cu ușurință pentru cei doi conectori proprietățile de comutativitate, asociativitate și distributivitate:

$$p \wedge q = q \wedge p \qquad p \vee q = q \vee p$$

$$p \wedge (q \wedge r) = (p \wedge q) \wedge r \qquad p \vee (q \vee r) = (p \vee q) \vee r$$

$$p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r) \qquad p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r).$$

Tot tablele arată că 0 este elementul neutru pentru disjuncție și 2 pentru conjuncție:

$$p \wedge 0 = 0 \wedge p = 0 \qquad p \wedge 2 = 2 \wedge p = p$$

$$p \vee 0 = 0 \vee p = p \qquad p \vee 2 = 2 \vee p = 2$$

Trebuie să ne obișnuim cu faptul că $1 \wedge p \neq p$ și $1 \wedge 2 = 2 \wedge 1 = 1$.

Pentru a defini implicația ne vom servi de formula din calculul bivalent

$$v(p \to q) = \min(1, 1 - v(p) + v(q))$$

și care devine în logica trivalentă

$$v(p \to q) = \min(2, 2 - v(p) + v(q)).$$

Obținem tabla

$p \underset{L}{\to} q$	0	1	2
0	2	2	2
1	1	2	2
2	0	1	2

Am notat această implicație $p \underset{L}{\to} q$ pentru că a fost utilizată de Lukasiewicz, unul din creatorii logicilor polivalente.

Dar în calculul bivalent mai avem o definiție a implicației:

$$v(p \to q) = \begin{cases} 1, \text{ dacă } v(p) \leqslant v(q), \\ v(q), \text{ dacă } v(p) > v(q), \end{cases}$$

ușor de verificat pe tabla de adevăr a implicației. Preluată în logica trivalentă (cu 2 în loc de 1), ea conduce la o altă implicație cu tabla următoare:

$p \underset{M}{\to} q$	0	1	2
0	2	2	2
1	0	2	2
2	0	1	2

Am notat această implicație $\underset{M}{\to}$ pentru că a fost utilizată de Moisil, creatorul unei algebre pentru logica polivalentă.

A treia definiție a implicației este dată de logica bivalentă de formula

$$p \to q = \neg p \vee q.$$

Dacă o preluăm pentru logica trivalentă, vom obține următoarea tablă:

$p \underset{K}{\to} q$	0	1	2
0	2	2	2
1	1	1	2
2	0	1	2

Am notat această implicație $\underset{K}{\to}$ pentru că a fost utilizată de Kleene, autorul unei versiuni a logicii trivalente.

Celor trei implicații le corespund table diferite pentru dubla implicație, care respectă cerința $p \leftrightarrow q = (p \to q) \wedge (q \to p)$

$p \underset{L}{\leftrightarrow} q$	0	1	2
0	2	1	0
1	1	2	1
2	0	1	2

$p \underset{M}{\leftrightarrow} q$	0	1	2
0	2	0	0
1	0	2	1
2	0	1	2

$p \underset{K}{\leftrightarrow} q$	0	1	2
0	2	1	0
1	1	1	0
2	0	1	2

Scanned with OKEN Scanner

O consecință a dispariției legii terțului exclus și a legii contradicției este că negația unui element nu poate servi drept complement al său. Complementul unui element x ar trebui să satisfacă condițiile

$$x + x' = 2, \qquad xx' = 0,$$

care conform tablei de mai înainte sînt satisfăcute de către $x' = \neg x$ pentru $x = 0, 2$, dar nu și pentru $x = 1$, deci $x' \neq \neg x$.

Părăsind cerințele unui complement, s-a introdus un operator $'$ numit *ciclare* (sau *alunecare*) definit după cum urmează :

x	x'	x''
0	1	2
1	2	0
2	0	1

7.2. DEZVOLTAREA FUNCȚIILOR

Pentru problema centrală a dezvoltării unei funcții a fost făcută încercarea de a găsi în logica trivalentă corespondenții minitermenilor, care permit în logica bivalentă scrierea unei funcții bine determinate atunci cînd este dată tabla ei de adevăr. Dar ciclurile nu pot servi la scrierea minitermenilor. Într-adevăr, în logica bivalentă minitermenul xy' ia valoarea 1 numai pentru interpretarea (1, 0) și 0 pentru celelalte, în timp ce în logica trivalentă xy' ia valoarea 1 și pentru (1, 0) și pentru (2, 0).

Pentru a rezolva această problemă se introduc noi operatori J_0, J_1, J_2 definiți după cum urmează :

x	$J_0(x)$	$J_1(x)$	$J_2(x)$
0	2	0	0
1	0	2	0
2	0	0	2

Definiția lui $J_i(x)$, unde $i = 0, 1, 2$, este astfel :

$$J_i(x) = \begin{cases} 2 & \text{pentru } x = i, \\ 0 & \text{altfel.} \end{cases}$$

În cazul unei funcții de o singură variabilă se poate scrie dezvoltarea

1) $f(x) = f(0)\, J_0(x) + f(1)\, J_1(x) + f(2) J_2(x)$.

Pentru două variabile se scrie

2) $f(x, y) = f(0, 0)\, J_0(x)\, J_0(y) + f(0, 1)\, J_0(x)\, J_1(y) +$
$+ f(0, 2)\, J_0(x)\, J_2(y) + f(1, 0)\, J_1(x)\, J_0(y) + f(1, 1)\, J_1(x)\, J_1(y) +$
$+ f(1, 2)\, J_1(x)\, J_2(y) + f(2, 0)\, J_2(x)\, J_0(y) +$
$+ f(2, 1)\, J_2(x) J_1(y) + f(2, 2)\, J_2(x)\, J_2(y)$.

Scanned with OKEN Scanner

Cu notațiile

$f(0,0)=d_0$, $f(0,1)=d_1$, $f(0,2)=d_2$, $f(1,0)=d_3$, $f(1,1)=d_4$

$f(1,2)=d_5$, $f(2,0)=d_6$, $f(2,1)=d_7$ și $f(2,2)=d_8$

și scriind m_i pentru minitermenul care corespunde lui $d_i=2$, formula (2) se poate scrie

$$3)\ f(x,y)=\sum_{0}^{8} d_i m_i.$$

Observăm că pentru o anumită evaluare a variabilelor, de exemplu $(1,2)$, toți minitermenii devin 0 în afară de $m_5=J_1(1)\,J_2(2)=2$ și deci $f(1,2)=d_5$. Cînd toți $d_i=0$, $f(x,y)=0$ și cînd $d_i=2$, $f(x,y)=2\sum_{i=0}^{8} m_i=2$.

Se deschide posibilitatea folosirii diagramelor Karnaugh, în mod analog cu cazul bivalent. Formăm diagramele

x \ y	$J_0(y)$	$J_1(y)$	$J_2(y)$
	0	1	2
$J_0(x)$ 0	m_0	m_1	m_2
$J_1(x)$ 1	m_3	m_4	m_5
$J_2(x)$ 2	m_6	m_7	m_8

în care vom înscrie coeficienții d_i corespunzători lui m_i. Se aplică apoi proprietatea de simplificare, pe care o oferă adunarea minitermenilor vecini cu un coeficient diferit de zero, la fel ca în modul de folosire a diagramelor Karnaugh în calculul bivalent.

Exemplu. Fie funcția definită de tabla de adevăr

x \ y	0	1	2
0	0	0	2
1	0	0	2
2	1	1	2

Conform formulei 3), ea se scrie dezvoltat sub forma

$$f(x,y)=2m_2+2m_5+1\cdot m_6+1\cdot m_7+2m_8.$$

Formăm harta Karnaugh indicînd două blocuri (fig. 7.1). Conform regulilor diagramelor Karnaugh putem să scriem în locul celor două blocuri funcțiile care corespund liniei 3 și coloanei 4:

$$f(x,y)=1\cdot J_2(x)+2\cdot J_2(y).$$

Într-adevăr, grupînd termenii celor două blocuri și aplicînd regulile ce definesc pe $J_i(x)$, obținem

$$2m_2+2m_5+2m_8=2J_0(x)\,J_2(y)+2J_1(x)\,J_2(y)+2J_2(x)J_2(y)=$$

$$=2J_2(y)\,(J_0(x)+J_1(x)+J_2(x))=2J_2(y)\cdot 2=J_2(y).$$

La fel, minitermenii din al doilea bloc se simplifică la $1\cdot J_2(y)$.

Scanned with OKEN Scanner

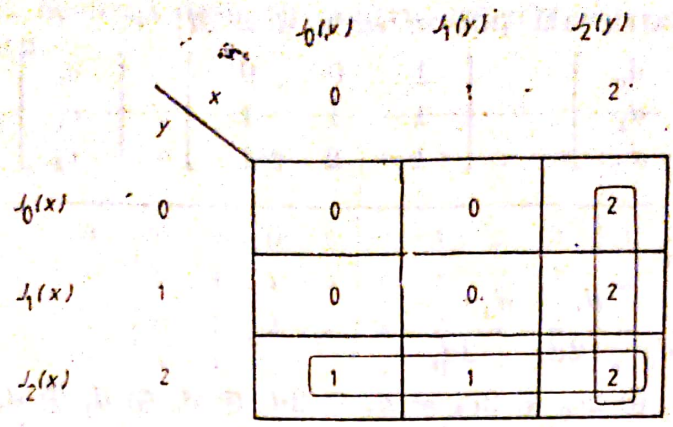


Fig. 7.1

Cunoscind avantajele calculului intr-un inel, următoarea problemă este de a determina dezvoltări ale funcțiilor de tip Reed-Müller.

Începem prin a extinde pentru logica trivalentă definirea unor operatori analogi cu operatorii $\oplus$ și $\cdot$. Vom apela la definițiile: din logica bivalentă:

$$v(x \oplus y) = [v(x) + v(y)] \bmod 2,$$

$$v(x \cdot y) = [v(x) \cdot v(y)] \bmod 2$$

și vom adopta pentru logica trivalentă

$$v(x \oplus y) = [v(x) + v(y)] \bmod 3,$$

$$v(x \cdot y) = [v(x) \cdot v(y)] \bmod 3,$$

ceea ce conduce la tablele de adevăr

$x \oplus y$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$x \cdot y$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Analog cu dezvoltările Reed-Müller din calculul bivalent vom compara în cazul unei singure variabile dezvoltarea căutată:

$$f(x) = c_0 \oplus c_1 x \oplus c_1 x^2$$

cu dezvoltarea

$$f(x) = f(0)\, J_0(x) + f(1)\, J_1(x) + f(2)\, J_2(x).$$

Vom ține de asemenea seama de relațiile

$$x \oplus x = 2x, \qquad x \cdot x = x^2,$$

$$x \oplus 2x = 0, \qquad x \cdot x^2 = x^3 = x.$$

$$x = 2x,$$

Coeficienții celor două dezvoltări sînt legați de relațiile

$$d_0 = f(0) = c_0$$

$$d_1 = f(1) = c_0 \oplus c_1 \oplus c_2$$

$$d_2 = f(2) = c_0 \oplus 2c_1 \oplus c_2.$$

Scanned with OKEN Scanner

În formă matriceală putem scrie $d = M_1 \cdot c$,

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}.$$

Din relațiile

$$c_0 = d_0$$
$$2\ c_2 = d_0 + d_1 + d_2$$
$$c_2 = 2d_0 + 2d_1 + 2d_2$$
$$c_1 = d_1 \oplus 2c_0 \oplus 2c_2 = d_1 \oplus 2d_0 \oplus d_0 \oplus d_1 \oplus d_2 = 2d_1 \oplus d_2$$

se obține matricea

$$c = M_1^{-1} \cdot d = N_1 d$$

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix}.$$

Pentru cazul a două variabile, dezvoltarea Reed-Müller va fi

$$f(x, y) = c_0 \oplus c_1 y \oplus c_2 y^2 \oplus c_3 x \oplus c_4 xy \oplus c_5 xy^2 \oplus c_6 x^2 \oplus c_7 x^2 y \oplus c_8 x^2 y^2$$

iar transformările vor fi date de matricele

$$c = M_2 d, \qquad d = N_2 c,$$

unde M_2 și N_2 au forma

$$M_2 = \begin{bmatrix} M_1 & 0 & 0 \\ M_1 & M_1 & M_1 \\ M_1 & 2M_1 & M_1 \end{bmatrix} \text{ și } N_2 = \begin{bmatrix} T_1 & 0 & 0 \\ 0 & 2T_1 & T_1 \\ 2T_1 & 2T_1 & 2T_1 \end{bmatrix}.$$

Observăm că spre deosebire de cazul binar $S_1 \neq T_1$.

Exemplu. Aplicînd procedeul descris la funcția din exemplul precedent, se obține o dezvoltare Reed-Müller:

$$f(x, y) = 1 \oplus 2y \oplus 2x \oplus xy \oplus xy^2 \oplus 2x^2 \oplus x^2 y^2.$$

Logică tetravalentă. Cu ajutorul calculului elaborat pentru logica trivalentă, putem să construim o logică tetravalentă, cu patru valori: adevărul (3) și falsul (0), între care intercalăm *adevărul contingent* (2) și *falsul contingent* (1). Există mai multe interpretări ale celor patru valori de adevăr, pe care le redăm în tabelul 7.1, unde S_1, S_2 sînt două sisteme logice diferite.

Tabelul 7.1

Valoare de adevăr	a	b	c	d
0	Necesarmente fals	Fals	Fals (demonstrabil)	Fals în S_1, S_2
1	Contingent fals	Probabil fals	Fals (arătat de fapte)	Fals în S_1, dar nu în S_2
2	Contingent adevărat	Probabil adevărat	Adevărat (arătat de fapte)	Adevărat în S_1, dar nu în S_2
3	Necesarmente adevărat	Adevărat	Adevărat (demonstrabil)	Adevărat în S_1, S_2

Scanned with OKEN Scanner

Recurgind la aceleași definiții pentru conectorii $\wedge$ și $\vee$, vom avea tabelele definitorii

	$\wedge$				$\vee$				$\neg$
	0	1	2	3	0	1	2	3	
0	0	0	0	0	0	1	2	3	3
1	0	1	1	1	1	1	2	3	2
2	0	1	2	2	2	2	2	3	1
3	0	1	2	3	3	3	3	3	0

Introducem de asemenea $J_0(x)$, $J_1(x)$, $J_2(x)$, $J_3(x)$ și ciclarea x':

x	0	1	2	3
$J_0(x)$	3	0	0	0
$J_1(x)$	0	3	0	0
$J_2(x)$	0	0	3	0
$J_3(x)$	0	0	0	3

x	x'	x''	x'''
0	1	2	3
1	2	3	0
2	3	0	1
3	0	1	2

Există o varietate de moduri de a defini implicația, care se dovedește a fi cel mai sensibil conector în construirea unui sistem logic. Cu elementele date, existența unei dezvoltări, unice a unei funcții logice, pornind de la tabla sa de adevăr, este asigurată.

7.3. ALGEBRE POST

Alegbra care corespunde logicii pozitive este o latice relativ pseudocomplementată, cea care corespunde logicii intuiționiste este o algebră pseudobooleană. Ce corespunde logicilor polivalente? După cum pornim de la o logică polivalentă Post sau de la una Lukasiewicz, dăm de o familie de algebre Post sau de algebre Lukasiewicz-Moisil. Există mai multe feluri de a introduce algebrele Post. Vom începe cu definirea algebrelor Post ca algebre pseudobooleene speciale, pentru a ne situa pe un teren explorat.

O algebră Post de ordinul 3 este o algebră $\mathscr{P} = \langle P, 2, \rightarrow, \vee, \wedge, \neg, D_1, D_2, e_0, e_1, e_2\rangle$, unde $\langle P, 2, \rightarrow, \vee, \wedge, \neg\rangle$ este o algebră pseudobooleană, în care 2 desemnează cel mai mare element, $\neg$, D_1, D_2 sînt operații cu un argument, iar $\rightarrow$, $\vee$, $\wedge$ operații cu două argumente, care satisfac următoarele ecuații:

p1 $D_i(a \vee b) = D_i(a) \vee D_i(b)$ $\quad i = 1, 2$

p2 $D_i(a \wedge b) = D_i(a) \wedge D_i(b)$ $\quad i = 1, 2$

p3 $D_1(a \rightarrow b) = D_1(a) \rightarrow D_1(b)$

$D_2(a \rightarrow b) = (D_1(a) \rightarrow D_1(b))(D_2(a) \rightarrow D_2(b))$

p4 $D_i(\neg a) = \neg D_1(a)$ $\quad i = 1, 2$

p5 $D_i(D_j(a)) = D_j(a)$ $\quad i, j = 1, 2$

p6 $D_i(e_j) = \begin{cases} 2 & i \leqslant j \\ \neg 2 & i > j \end{cases}$

Scanned with OKEN Scanner

p7 $a = e_1 D_1(a) + e_2 D_2(a)$

p8 $D_1(a) \vee \neg D_1(a) = 2.$

Pentru ușurarea scrierii formulelor vom scrie uneori xy în loc de $x \wedge y$. De exemplu p7 este forma prescurtată a lui

$$a = (D_1(a) \wedge e_1) \vee (D_2(a) \wedge e_2)$$

Într-o algebră Post de ordinul 3 au loc următoarele proprietăți :

p1′ $\langle P, 2, \wedge, \vee \rangle$ este o latice distributivă cu elementul unitate 2 și cu elementul $0 = \neg 2$.

p2′ $0 = e_0 \leqslant e_1 \leqslant e_2 \leqslant 2.$

Demonstrații. Folosim p6 și p7 și p1′ pentru a arăta că e_i $(i = 0, 1, 2)$ formează un lanț : $e_0 = 0$, $a_2 = e_1 \vee e_2$, $e_2 \geqslant e_1$ și $e_1 \geqslant e_0$, deci $0 = e_0 \leqslant e_1 \leqslant e_2$. De aici deducem $e_i e_j = e_i$ pentru $i \leqslant j$, $i, j = 0, 1, 2$, și deci $ae_2 = a$ pentru orice $a \in P$. Înseamnă că $a \leqslant e_2$ pentru orice a și deci $e_2 = 2$.

p3′ $D_i(a) \leqslant D_j(a)$ pentru $j \leqslant i$, $i, j = 1, 2$, $a \in P$.

p4′ Dacă $a \leqslant b$, atunci $D_i(a) \leqslant D_i(b)$ pentru $i = 1,2$ și orice $a, b \in P$.

p5′ $D_2(a) \leqslant a \leqslant D_1(a).$

p6′ $e_0 \to a = e_2$, $e_1 \to a = D_1(a)$, $e_2 \to a = a$ pentru orice $a \in P$.

Demonstrație. Avem $e_0 \to a = 0 \to a = 2 = e_2$ și $e_2 \to a = 2 \to a = a$, deci

$$e_1 \to a = D_1(e_1 \to a)\, e_1 \vee D_2(e_1 \to a) e_2 =$$
$$= (D_1(e_1) \to D_1(a)) e_1 \vee (D_1(e_1) \to D_1(a))\,(D_2(e_1) \to D_2(a))\, e_2 =$$
$$= (e_2 \to D_1(a))\, e_1 \vee (e_2 \to D_1(a))\,(e_0 \to D_2(a))\, e_2 = D_1(a)$$

p7′ $a \to e_0 = \neg a$, $a \to e_2 = e_2$ pentru orice $a \in P$

p8′ $a \to e_1 = e_1 \vee \neg D_2(a)$ pentru orice $a \in P$

p9′ $e_i \to e_j = \begin{cases} e_2, & i \leqslant j, \\ e_j, & i > j, \end{cases}$ $e_i \neq e_j$ și $i, j = 0, 1, 2.$

p10′ În orice algebră Post nedegenerată, dacă $i \neq j$,

$$e_i \neq e_j \ (i, j = 0, 1, 2).$$

Studiul tuturor elementelor de forma $D_i(a)$, cu $i = 1, 2$ și $a \in P$, ne conduce la următoarea proprietate importantă :

1) Într-o algebră Post, mulțimea B a tuturor elementelor de forma $D_i(a)$, $i = 1, 2$ și $a \in P$, este mulțimea tuturor elementelor complementate în laticea distributivă $\langle P, 2, \vee, \wedge \rangle$.

2) Mulțimea B este închisă față de operațiile $\to$, $\vee$, $\wedge$, $\neg$ și algebra $\mathscr{B} = \langle B, 2, \to, \vee, \wedge, \neg \rangle$ este o algebră booleană.

Demonstrație. Pe de o parte $a \wedge \neg a = 0$ și $D_i(a) \wedge \neg D_i(a) = 0$ cu $i = 1, 2$, pentru că $\langle P, 2, \to, \vee, \wedge, \neg \rangle$ este o algebră pseudobooleană.

Scanned with OKEN Scanner

Pe de altă parte $D_i(a) \vee \neg D_i(a) = 2$, $i = 1, 2$. D fiecare element al lui B este complementat : $\neg D(a_i)$ este complem ul lui $D_i(a)$, $i = 1, 2$ și $a \in P$.

Fie a un element oarecare complementa Complementul său este $\neg a$, după cum rezultă din relațiile $a \wedge b = 0$ $a \vee b = 2$, $b \leqslant a \to 0 = \neg a$ și $2 = a \vee b \leqslant a \vee \neg a$. Dar

$$\neg a = e_1 D_1(\neg a) \vee e_2 D_2(\neg a)' = \neg D_1(a)(e_1 \vee e_2) = \neg D_i(a).$$

Cum a și $D_1(a)$ sînt complementate, avem

$$a = \neg \neg a = \neg \neg D_1(a_1) = D_1(a).$$

Astfel orice element complementat a este de forma $D_1(a)$ și aparține lui B.

O altă axiomatizare a algebrelor Post, mai frecvent citată, diferă de cea introdusă anterior prin faptul că pornește de la o latice distributivă cu element zero și element unitate, iar în locul negației folosește operația C, definită ca $Ca = \neg D_1(a)$ pentru orice $a \in P$.

Vom trata iarăși cazul algebrelor de ordinul trei.

Fie $\mathscr{P}$ o algebră Post de ordinul 3. Introducem operația C prin

$$Ca = \neg D_1(a) \quad \text{pentru orice } a \in P.$$

Algebra $\mathscr{P}^* = \langle P, 2, \vee, \wedge, C, D_1, D_2, e_0, e_1, e_2 \rangle$ are următoarele proprietăți :

p1* $\langle P, 2, \vee, \wedge \rangle$ este o latice distributivă cu elementul unitate și elementul zero, $0 = e_0$, $2 = e_2$

p2* $e_0 \leqslant e_1 \leqslant e_2$,

p3* $D_1(a) \wedge C(a) = 0$, $D_1(a) \vee C(a) = 2$,

p4* $D_i(a) \wedge D_j(a) = D_j(a)$, $\quad i \leqslant j$, $\quad i = 1, 2$

p5* $D_i(a \vee b) = D_i(a) \vee D_i(b)$ și

$D_i(a \wedge b) = D_i(a) \wedge D_i(b)$, $\quad i = 1, 2$

p6* $D_i(D_j(a)) = D_j(a)$, $\quad i, j = 1, 2$

p7* $D_i(e_j) = \begin{cases} e_2, & i \leqslant j, \\ e_0, & i > j, \end{cases}$ $\quad i = 1, 2$; $j = 0, 1, 2$

p8* $a = e_1 D_1(a) \vee e_2 D_2(a)$,

unde a, b sînt elemente oarecare din P.

Se observă că p5* este identic cu p1 și p2; p6*, p7*, p8* sînt identice cu p5, p6, p7, iar p1*, p2*, p3* și p4* rezultă imediat din p1′ — p10′.

Pentru a demonstra echivalența celor două moduri de a introduce algebrele Post, demonstrăm propoziția următoare.

Fie $\mathscr{P} = \langle P, 2, \vee, \wedge, C, D_1, D_2, e_0, e_1, e_2\rangle$ o algebră care satisface p1* – p8*. Fie $\to$ și $\neg$ două operații definite de ecuațiile

$$a \to b = (CD_1(a) \vee D_1(b))\, e_1 \vee (CD_1(a) \vee D_1(b))\,(CD_2(a) \vee D_2(b))\, e_2,$$

$$\neg a = a \to e_0$$

pentru orice $a, b \in P$. Atunci algebra $\mathscr{P} = \langle P, 2, \to \vee, \wedge, \neg, D_1, D_2, e_0, e_1, e_2\rangle$ este o algebră Post de ordinul 3.

Demonstrația este mai laborioasă. Nu este greu să se arate că :

a) $CD_i(a)$ este complementul lui $D_i(a)$, $i = 1, 2,$;

b) $D_j(a) \wedge CD_i(a) = e_0$, $\qquad i \leqslant j,\ i, j = 1, 2$;

c) $D_j(CD_i(a)) = CD_i(a)$, $\qquad i = 1, 2.$

Vom arăta că operația $\to$ satisface cerințele unui pseudocomplement relativ :

$a \wedge x \leqslant b$ dacă și numai dacă $x \leqslant a \to b$ pentru toți $x, a, b \in P$.

Demonstrăm mai întîi că pentru toți $a, b \in P$, $a \wedge (a \to b) \leqslant b$. Avem

$$a \wedge (a \to b) = (D_1(a)\, e_1 \vee D_2(a)\, e_2)[(CD_1(a) \vee D_1(b))e_1 \vee$$

$$\vee\, (CD_1(a) \vee D_1(b))\,(CD_2(a) \vee D_2(b))e_2]$$

din p8* și definiția lui $\to$. Din p2* și distributivitate, folosind b) :

$$a \wedge (a \to b) \leqslant D_1(b)e_1 \vee D_2(b)\, e_2 = b.$$

Cu ajutorul acestei relații se arată că $x \leqslant a \to b$ implică

$$a \wedge x \leqslant a \wedge (a \to b) \leqslant b.$$

Reciproc, dacă $a \wedge x \leqslant b$, atunci

$$D_i(a) \wedge D_i(x) \leqslant D_i(b), \qquad i = 1, 2,$$

din p1* și p5*. În continuare pentru $i = 1, 2$, avem

$$CD_i(a) \vee (D_i(a) \wedge D_i(x)) \leqslant CD_i(a) \vee D_i(b),$$

$$(CD_i(a) \wedge D_i(a)) \wedge (CD_i(a) \vee D_i(x)) \leqslant CD_i(a) \vee D_i(b),$$

$$D_i(x) \leqslant CD_i(a) \vee D_i(b),$$

unde am folosit c). Dar

$$x = D_i(x)\, e_1 \vee D_2(x)e_2 \leqslant D_1(x)e_1 \vee D_1(x)D_2(x)e_2 \leqslant$$

$$\leqslant (CD_1(a) \vee D_1(b))e_2 \vee (CD_1(a) \vee D_1(b))(CD_2(a) \vee D_2(b))e_2 = a \to b,$$

ceea ce completează demonstrația.

Enunțăm acum pentru cazul general cele două definiții ale algebrelor Post :

A) Se numește o *algebră Post de ordinul n*

$$\mathscr{P} = \langle P, U, \to, \vee, \wedge, \neg, D_1, \ldots, D_{n-1}, e_0, e_1, \ldots, e_{n-1}\rangle, \quad n \geqslant 2,$$

unde $U, e_0, \ldots, e_{n-1}$ sînt operații cu 0 argumente, $\neg, D_1, \ldots, D_{n-1}$ cu 1 argument și $\to, \vee, \wedge$ cu 2 argumente, definite de următoarele ecuații:

P0 $\langle P, U, \to, \vee, \wedge, \neg \rangle$ este o algebră pseudobooleană

P1 $D_i(a \vee b) = D_i(a) \vee D_i(b), \qquad i = 1, 2, \ldots, n-1;$

P2 $D_i(a \wedge b) = D_i(a) \wedge D_i(b), \qquad i = 1, 2, \ldots, n-1;$

P3 $D_i(a \to b) = (D_1(a) \to D_1(b)) \ldots (D_i(a) \to D_i(b)),$

P4 $D_i(\neg a) = \neg D_i(a), \qquad i = 1, 2, \ldots, n-1;$

P5 $D_i(D_j(a)) = D_j(a), \qquad i, j = 1, 2, \ldots, n-1;$

P6 $D_i(e_j) = \begin{cases} U, & i \leqslant j, \\ \neg U, & i > j, \end{cases} \qquad \begin{matrix} i = 1, 2, \ldots, n-1, \\ j = 0, 1, \ldots, n-1; \end{matrix}$

P7 $a = D_1(a)\, e_1 \vee D_2(a) e_2 \vee \ldots \vee D_{n-1}(a)\, e_{n-1},$

P8 $D_1(a) \vee \neg D_1(a) = U,$

unde $a, b \in P$.

B) Se numește *o algebră Post de ordinul n*

$$\mathscr{P}^* = \langle P, U, \vee, \wedge, C, D_1, \ldots, D_{n-1}, e_0, e_1, \ldots, e_{n-1} \rangle$$

care satisfac proprietățile

P1* $\langle P, U, \vee, \wedge \rangle$ este o latice distributivă cu elementul unitate $U = e_{n-1}$ și elementul zero $0 = e_0$;

P2* $e_0 \leqslant e_1 \leqslant \ldots \leqslant e_{n-1};$

P3* $D_1(a) \wedge C(a) = 0, \qquad D_1(a) \vee C(a) = U;$

P4* $D_i(a) \wedge D_j(a) = D_j(a), \qquad i \leqslant j,\ i, j = 1, 2; \ldots, n-1;$

P5* $D_i(a \vee b) = D_i(a) \vee D_i(b), \quad D_i(a \wedge b) = D_i(a) \wedge D_i(b),$

$i, j = 1, 2, \ldots, n-1;$

P6* $D_i(D_j(a)) = D_j(a), \qquad i, j = 1, 2, \ldots, n-1;$

P7* $D_i(e_j) = \begin{cases} e_{n-1}, & i \leqslant j,\ i = 1, \ldots, n-1, j = 0, \ldots, n-1; \\ e_0, & i > j, \end{cases}$

P8* $a = e_1 D_1(a) \vee \ldots \vee e_{n-1} D_{n-1}(a),$

unde $a, b \in P$.

Definiția A) are avantajul de a sublinia încadrarea algebrelor Post în algebrele pseudobooleene, iar B) are avantajul de a utiliza operația C în loc de $\to$ și $\neg$.

Algebra Post cu n elemente este un exemplu de algebră Post de ordinul n. Ea este $\mathscr{P}_n = \langle P_n, U, \to, \vee, \wedge, \neg, D_1, \ldots, D_{n-1}, e_0, e_1, \ldots, e_{n-1} \rangle$. Mulțimea P_n este $\{e_0, \ldots, e_{n-1}\}$. Au loc următoarele relații:

1) $e_{m-1} = U;$

2) $e_i \to e_j = \begin{cases} U, & i \leqslant j, \\ e_j, & i > j; \end{cases}$

3) $e_i \vee e_j = e_k$, unde $k = \max(i, j)$;

4) $e_i \wedge e_j = e_k$, unde $k = \min(i, j)$;

5) $\neg e_i = e_i \to e_0$,

6) $D_i(e_j) = \begin{cases} U, & i \leqslant j, \\ e_0, & i > j. \end{cases}$

Notăm $J_i(x) = D_i(x) \wedge D_{i+1}(x)$, $i = 0, \ldots, n-1$. Se verifică

$$J_i(e_j) = \begin{cases} e_{n-1}, & i = j, \\ e_0, & i \neq j. \end{cases}$$

Prin algebra Post cu n elemente se face legătura cu 7.1 și 7.2 și se regăsesc relațiile stabilite pentru $n = 3$.

7.4. ALGEBRA LUKASIEWICZ-MOISIL (LM)

Pentru logica trivalentă a lui Lukasiewicz, Gr. C. Moisil a construit o algebră $\mathscr{L} = \langle L, N, \mu, 0, 2 \rangle$ cu următoarele axiome și proprietăți:

1) L este latice distributivă mărginită.

2) N este negația, definită de

$$N(x \wedge y) = Nx \vee Ny,\ N(x \vee y) = Nx \wedge Ny,\ NNx = 0.$$

3) L are un endomorfism $x \to \mu x$ (μx citim x este posibil), cu proprietățile

$$\mu(x \wedge y) = \mu x \wedge \mu y,\ \ \mu(x \vee y) = \mu x \vee \mu y,$$

$$x \leqslant \mu x,\ \ \mu\mu x = \mu x,\ \ \mu 0 = 0.$$

4) Dubla imposibilitate

$$N\mu N\mu = \mu.$$

5) Cvartul exclus

$$N\mu Nx \vee (\mu x \wedge \mu Nx) \vee N\mu x = 2 \text{ sau}$$

$$\mu Nx \wedge (N\mu x \vee N\mu Nx) \wedge \mu x = 0.$$

Acesta poate fi enunțat și sub forma *principiului modal al contradicției* și al terțului exclus

$$x \wedge N\mu x = 0, \qquad x \vee \mu Nx = 2.$$

6)
$$x \wedge Nx = \mu x \wedge Nx.$$

Scanned with OKEN Scanner

Observăm avantajul considerabil al organizării algebrei LM prin introducerea unui singur operator unar μ. Acesta nu se comportă parțial ca „posibilul" din logica modală și care, operînd împreună cu N, generează ceilalți operatori modali :

$$\nu(x) = N\mu N(x), \quad \gamma(x) = \mu N(x), \quad \eta(x) = N\mu(x).$$

În logica modală vom regăsi pe ν sub forma $\Box$ (necesarul), iar pe μ sub forma $\Diamond$ (posibilul). Cu toate acestea, o diferență esențială și anume că în logica modală $\mu(x \wedge y) \neq \mu(x) \wedge \mu(y)$ împiedică asimilarea logicii polivalente cu logica modală.

Aceste definiții conduc la tabloul compunerii operatorilor modali :

	I	N	ν	μ	γ	η
I	I	N	ν	μ	γ	η
N	N	I	γ	η	ν	μ
ν	ν	η	ν	μ	γ	η
μ	μ	γ	ν	μ	γ	η
γ	γ	μ	γ	η	γ	μ
η	η	ν	γ	η	γ	μ

Interpretarea în logica trivalentă este următoarea :

x	$=$ 0	1	2
$\nu(x)$	0	0	2
$\mu(x)$	0	2	2
$\gamma(x)$	2	2	0
$\eta(x)$	2	0	0

Observăm că $J_2(x) = \nu(x)$, $J_0(x) = \eta(x)$, iar $J_1(x)$ se poate scrie $J_1(x) = \mu(x) \wedge \gamma(x)$.

Această corespondență ne permite să putem compara dezvoltarea obținută de Moisil pentru funcțiile de una și două variabile cu formulele dezvoltării cu utilizarea lui J_0, J_1, J_2, 1) și 2) :

$$f(x) = f(0)\,\eta(x) + f(1)\,\mu(x)\,\gamma(x) + f(2)\,\nu(x) \quad \text{și}$$

$$\begin{aligned} f(x, y) = {} & f(0,0)\,\eta(x)\,\eta(y) + f(0,1)\,\eta(x)\,\mu(y)\,\gamma(y) + \\ & + f(0,2)\,\eta(x)\,\nu(y) + f(1,0)\,\mu(x)\,\gamma(x)\,\eta(y) + \\ & + f(1,1)\,\mu(x)\,\gamma(x)\,\mu(y)\,\gamma(y) + f(1,2)\,\mu(x)\,\gamma(x)\,\nu(y) + \\ & + f(2,0)\,\nu(x)\,\eta(y) + f(2,1)\,\nu(x)\,\mu(y)\,\gamma(y) + \\ & + f(2,2)\,\nu(x)\,\nu(y). \end{aligned}$$

Alte relații remarcabile sînt cele (echivalente) ale principiului modal al contradicției :

a) $x \wedge N\mu x = 0$,

b) $Nx \wedge N\mu Nx = 0$,

c) $\mu x \wedge N\mu x = 0$,

d) $\mu Nx \wedge N\mu Nx = 0$,

e) $N\mu x \wedge N\mu Nx = 0$

și cele (echivalente) ale principiului terțului exclus

a′) $Nx \vee \mu x = 1$,

b′) $x \vee \mu Nx = 1$,

c′) $N\mu x \vee \mu x = 1$,

d′) $N\mu Nx \vee \mu Nx = 1$,

e′) $\mu x \vee \mu Nx = 1$,

precum și formele echivalente cu axioma 6 :

a″) $x \wedge Nx = \mu x \wedge Nx$,

b″) $x \wedge Nx = x \wedge \mu Nx$,

c″) $x \vee Nx = x \vee N\mu x$,

d″) $x \vee Nx = Nx \vee N\mu Nx$.

Observăm că punctul de plecare al acestei algebre este o latice distributivă L, care are proprietățile :

$$NNx = x, \quad N(x \vee y) = Nx \wedge Ny.$$

Moisil a numit această latice și latice De Morgan. În literatură ea mai este întîlnită astăzi sub denumirea de *algebră cvasibooleană* : o algebră $\langle L, 1, \vee, \wedge, \neg \rangle$, unde $\langle L, 1, \vee, \wedge \rangle$, este o latice distributivă cu 1 și unde $\neg$ satisface

$$\neg\neg a = a, \quad \neg(a \vee b) = \neg a \vee \neg b.$$

Într-o algebră cvasibooleană (latice De Morgan) există și elementul 0 : $\neg 1 = 0$, $\neg 0 = 1$ și are loc relația

$$\neg(a \wedge b) = \neg a \vee \neg b$$

Regrupînd proprietățile stabilite de Moisil și utilizînd notațiile $\neg$ pentru N, D_1 pentru μ, D_2 pentru ν, 2 sau U pentru 1, se obține următoarea axiomatizare a unei *algebre Lukasiewicz-Moisil de ordinul 3*, $\mathscr{L} = \langle L, 2, \vee, \wedge, D_1, D_2, \neg \rangle$:

l1 L este o algebră De Morgan cu operațiile

$$\wedge, \ \vee, \ \neg, \ 0, \ 2$$

l2 $D_i(a \vee b) = D_i(a) \vee D_i(b)$, $D_i(a \wedge b) = D_i(a)\, D_i(b)$, $i = 1, 2$;

l3 $D_i(a)\, D_j(a) = D_j(a)$, $i, j = 1, 2$ și $i \leqslant j$;

l4 $D_i(a) \vee \neg D_i(a) = 1$, $D_i(a) \cdot \neg D_i(a) = 0$, $i = 1, 2$;

l5 $D_i(\neg a) = \neg D_i(a)$, $i = 1, 2$;

l6 $D_i(D_j(a)) = D_j(a)$, $i, j = 1, 2$;

l7 $a \vee D_1(a) = D_1(a)$, $a \cdot D_2(a) = D_2(a)$:

Scanned with OKEN Scanner

I8 $D_i(0) = 0, \quad D_i(1) = 1, \qquad i = 1, 2;$

I9 $\neg a \wedge D_2(a) = 0, \qquad \neg a \vee D_1(a) = 1;$

I10 $b(a \vee \neg D_1(a) \vee D_2(b)) = b.$

Această formă a axiomelor pentru logica trivalentă permite o generalizare imediată pentru algebrele Lukasiewicz-Moisil $\mathcal{L} = \langle L, U, \vee, \wedge, D_1, \ldots, D_{n-1}, \neg \rangle$ de ordinul n, caracterizate de:

L1 L este o algebră De Morgan cu operațiile $\wedge$, $\vee$, $\neg$, 0, U

L2 $D_i(a \vee b) = D_i(a) \vee D_i(b), \; D_i(a \wedge b) = D_i(a)\, D_i(b), \; i = 1, \ldots, n-1;$

L3 $D_i(a)\, D_j(a) = D_j(a), \qquad i \leqslant j, \; i, j = 1, \ldots, n-1;$

L4 $D_i(a) \vee \neg D_i(a) = U, \qquad D_i(a) \cdot \neg D_i(a) = 0, \; i = 1, \ldots, n-1;$

L5 $D_i(\neg a) = \neg D_{n-1}(a), \qquad i = 1, \ldots, n-1;$

L6 $D_i(D_j(a)) = D_j(a), \qquad i, j = 1, \ldots, n-1;$

L7 $a \vee D_1(a) = D_1(a), \qquad a \cdot D_{n-1}(a) = D_{n-1}(a);$

L8 $D_1(0) = 0, \; D_i(1) = 1, \qquad i = 1, 2, \ldots, n-1;$

L9 $\neg a \cdot D_{n-1}(a) = 0, \qquad \neg a \vee D_1(a) = U;$

L10 $b(a \vee \neg D_i(a) \vee D_{i+1}(b)) = b, \qquad i = 1, \ldots, n-2.$

Legătura între algebrele Post și algebrele Lukasiewicz-Moisil. Fie o algebră Post $\langle P, U, \vee, \wedge, D_1, \ldots, D_{n-1}, e_0, e_1, \ldots, e_{n-1} \rangle$ de ordinul n. Punînd

$$\neg a = \bigvee_{i=1}^{n-1} \neg D_{n-i}(a)\, e_i, \text{ unde } \neg D_{n-i}(a) = C(D_{n-i}(a)),$$

se demonstrează că $\langle P, U, \vee, \wedge, D_1, \ldots, D_{n-1}, \neg \rangle$ este o algebră Lukasiewicz-Moisil de ordinul n, întrucît au loc L1 — L10.

Reciproc, fie o algebră Lukasiewicz-Moisil $\langle L, U, \vee, \wedge, D_1, \ldots, D_{n-1}, \neg \rangle$, care are elemente $e_0, \ldots, e_{n-1}$ astfel ca

$$D_i(e_j) = U, \qquad i \leqslant j, \; i, j = 1, 2, \ldots, n-1;$$

$$D_i(e_j) = 0, \; j < i, \; i = 1, \ldots, n-1, \; j = 0, \ldots, n-1;$$

atunci $\langle L, U, \vee, \wedge, C, D_1, \ldots, D_{n-1}, e_1, \ldots, e_{n-1} \rangle$ este o algebră Post de ordinul n, unde C este definit de $C(a) = \neg D_1(a)$ pentru $a \in L$, întrucît axiomele $P1^* - P8^*$ se verifică.

Scanned with OKEN Scanner

4.5. SISTEME LOGICE POLIVALENTE

Completăm tabloul schițat în 7.1 cu privire la cele mai cunoscute și folosite logici polivalente, prezentîndu-le pe scurt elementele definitorii (semantică și axiomele) :

a) **Sistemul trivalent al lui Lukasiewicz** (L_3). Valori de adevăr : A (adevărat), I (nedeterminat sau neutru), F (fals). Negația și implicația pot fi luate conectori primitivi. În acest caz

$$p \vee q = (p \to q) \to q$$

$$p \wedge q = \neg(\neg p \vee \neg q)$$

$$p \leftrightarrow q = (p \to q) \wedge (q \to p)$$

Tablele de adevăr sînt :

p \ q	$p \wedge q$			$p \vee q$			$p \to q$			$p \leftrightarrow q$			$\neg p$
	A	I	F	A	I	F	A	I	F	A	I	F	
A	A	I	F	A	A	A	A	I	F	A	I	F	F
I	I	I	F	A	I	I	A	A	I	I	A	I	I
F	F	F	F	A	I	F	A	A	A	F	I	A	A

(A, I, F apar la diferiți autori ca 2, 1, 0 sau 1, 1/2, 0).

Axiomele :

1) $P \to (Q \to P)$

2) $(P \to Q) \to [(Q \to R) \to (P \to R)]$

3) $(\neg Q \to \neg P) \to (P \to Q)$

4) $[(P \to \neg P) \to P] \to P$

b) **Logica n-valentă a lui Lukasiewicz** (L_n)

Valorile de adevăr sînt

$$0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-2}{n-1}, \frac{n-1}{n-1} = 1$$

Tablele de adevăr se alcătuiesc după regulile :

$$|\neg p| = 1 - |p|$$

$$|p \wedge q| = \min |p|, |q|$$

$$|p \vee q| = \max |p|, |q|$$

$$|p \to q| = \begin{cases} 1, & |p| \leqslant |q| \\ 1 - |p| + |q|, & |p| > |q| \end{cases}$$

$$|p \leftrightarrow q| = |(p \to q) \wedge (q \to p)|$$

Scanned with OKEN Scanner

Axiomele :

1) $P \to (Q \to P)$

2) $(P \to Q) \to [(Q \to R) \to (P \to R)]$

3) $(\neg Q \to \neg P) \to (P \to Q)$

4) $[(P \to Q) \to P] \to [(Q \to P) \to P]$

c) **Sistemul trivalent al lui Bochvar (B_3).** Valorile de adevăr sînt *A*, *I*, *F*.

Tablele de adevăr :

p \ q	$p \wedge q$			$p \vee q$			$p \to q$			$p \leftrightarrow q$			$\neg p$
	A	*I*	*F*	*A*	*I*	*F*	*A*	*I*	*F*	*A*	*I*	*F*	
A	*A*	*I*	*F*	*A*	*I*	*A*	*A*	*I*	*F*	*A*	*I*	*F*	*F*
I	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*I*
F	*F*	*I*	*F*	*A*	*I*	*F*	*A*	*I*	*A*	*F*	*I*	*A*	*A*

Fără axiome.

Conectorii primitivi sînt $\neg$ și $\wedge$. Ceilalți conectori sînt introduși prin

$$p \vee q = \neg(\neg p \wedge \neg q)$$
$$p \to q = \neg(p \wedge \neg q)$$
$$p \leftrightarrow q = [(p \to q) \wedge (q \to p)]$$

Observăm că *I* are la Bochvar sensul de „nedecidabil". Valoarea *I* nu este atît o valoare intermediară între *A* și *F*, ci reprezintă un element paradoxal, ceea ce face ca într-o conjuncție, este suficient ca unul din termeni să aibă valoarea *I*, ca întreaga conjuncție să fie *I*.

Bochvar introduce pe lîngă „aserțiunea internă" a sistemului de mai sus o „aserțiune externă" cu un operator A, care nu ia decît valorile *A*, *F*.

d) **Sistemul logic polivalent a lui Bochvar (B_n).** Valorile de adevăr :

$$0 = \frac{0}{n-1}, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, \frac{n-1}{n-1} = 1$$

Reguli de adevăr :

$$|\neg p| = 1 - |p|$$

$$|p \wedge q| = \begin{cases} |p| \times |q|, & \text{ambele } p, q \text{ aparțin mulțimii } \{0, 1\}, \\ Z(n) & \text{altfel.} \end{cases}$$

$$|p \vee q| = \begin{cases} \min [1, |p| + |q|, & \text{ambele } p, q \text{ aparțin mulțimii } \{0, 1\}, \\ Z(n) & \text{altfel.} \end{cases}$$

$$|p \to q| = \begin{cases} \min 1, 1 - |p| + |q|, & \text{ambele } p, q \text{ aparțin mulțimii } \{0, 1\}, \\ Z(n) & \text{altfel.} \end{cases}$$

$$|p \leftrightarrow q| = \begin{cases} 1 - |p| - |q| + 2|p| \times |q|, & \text{ambele } p, q \text{ aparțin mulțimii } \{0, 1\}, \\ Z(n) & \text{altfel.} \end{cases}$$

unde $Z(n) = \begin{cases} \frac{1}{2}, & n \text{ impar}, \\ \frac{n-2}{2(n-1)}, & n \text{ par}. \end{cases}$

e) **Sistemul logic trivalent al lui Post (P_3).** Valorile de adevăr : 1, 2, 3. Tablele de adevăr sînt :

p \ q	$p \wedge q$			$p \vee q$			$p \to q$			$p \leftrightarrow q$			$\neg p$
	1	2	3	1	2	3	1	2	3	1	2	3	2
1	3	3	2	1	1	1	1	2	2	3	3	3	2
2	3	1	2	1	2	2	1	2	3	3	1	2	3
3	2	2	2	1	2	3	1	1	1	3	2	3	1

f) **Sistemul logic polivalent al lui Post (P_n).** Negația este definită de tabla de adevăr

p	$\neg p$
1	2
2	3
3	4
.	.
.	.
.	.
$n-1$	n

Regulile de adevăr sînt :

$$|\neg p| = |p| + 1 (\text{mod } n)$$

$$|p \vee q| = \min [|p|, |q|[$$

$$|p \wedge q| = |\neg(\neg p \vee \neg q)|$$

$$|p \to q| = \neg p \vee q$$

$$|p \leftrightarrow q| = [(p \to q) \wedge (q \to p)]$$

7.6. NOTĂ ISTORICĂ

Problematica „legii bivalenței" a fost dezbătută în antichitate. Aristotel s-a întrebat ce statut de adevăr au propozițiile privind viitorul contingent ca de exemplu : „mîine se va da o bătălie navală", care nu este nici adevărată, nici falsă, deși poate fi și adevărată și falsă. Stoicii, care erau determiniști, au susținut legea bivalenței, printre ei distingîndu-se în această direcție Chrysip. Școala epicureiană nedeterministă a respins principiul bivalenței.

În Evul mediu, Duns Scotus și William de Ockham au distins propozițiile neutre de cele adevărate și de cele false.

Scanned with OKEN Scanner

Cele mai importante încercări de a elabora logici polivalente, după ce logica bivalentă simbolică și formalizată cunoscuse progrese însemnate, se datorează lui Hugh Mac Coll (1837—1909), lui Charles Sanders Peirce (1839—1914) și lui Nicolai Vasiliev (1880—1940).

În 1920 Lukasiewicz publică sistemul său trivalent, iar în 1921 Post prezintă sistemul său (inițial pentru n valori). Școala poloneză a studiat fără întrerupere logicile neclasice de la Lukasiewicz încoace, ilustrîndu-se prin contribuțiile lui A. Tarski, J. Slupecki, B. Sobocinski și Elena Rasiowa astăzi. Apariția logicilor polivalente în același timp cu dezvoltarea logicii intuiționiste — negarea terțului exclus fiind preocuparea comună a celor două școli — a făcut ca mulți matematicieni și logicieni să contribuie în ambele direcții : A. Heyting, K. Gödel, Kolmogorov. Sistemele Kleene și Bochvar datează din deceniul 1930. Încercările numeroase de a studia logicile modale cu ajutorul logicilor polivalente au fost părăsite, după apariția unei teoreme de imposibilitate datorită lui J. Dergundji (1940), conform căreia logicile modale ale lui Lewis nu admit matrice finită caracteristică, adică algebre finite în care să fie modelate.

Studiul algebric al logicilor polivalente s-a făcut la două decenii de la apariția lor. Algebrele Post au fost elaborate de Rosenbloom în 1942 și studiate mai amănunțit de Epstein și Traczyk după 1960. În 1940 și 1941 Gr. C. Moisil a introdus algebrele pe care le-a numit algebre Lukasiewicz, dar care în literatură apar după 1960 și ca algebre Lukasiewicz-Moisil, drept recunoaștere a operei de pionierat a matematicianului român. În tratatele de logică sau de algebră (Rasiowa, Balbes și Dwinger, Rescher) această recunoaștere este explicită. Adîncirea rezultatelor sale a fost făcută de școala de la București, ca și de A. Monteiro, L. Monteiro, R. Cignoli și alții. Laticele (algebrele) De Morgan sînt studiate și numite astfel de Gr. C. Moisil în 1935.

Logicile polivalente au aplicații în calculul circuitelor electronice și în fizica modernă. Încă din 1946 H. Reichenbach a construit o logică cuantică trivalentă.

7.7. EXERCIȚII ȘI PROBLEME

1. Să se arate că dacă $\mathscr{P} = \langle P, 2, \rightarrow, \vee, \wedge, \neg, D_1, e_0, e_1 \rangle$ este o algebră Post de ordinul 2, atunci $e_1 = 2$, $e_0 = \neg 2 = 0$, $D_1(a) = a$ pentru orice $a \in P$ și $\langle P, 2, \rightarrow, \vee, \wedge, \neg \rangle$ este o algebră booleană, unde 2 este cel mai mare element.

2. Să se demonstreze p1′, p3′ — p5′, p7′ — p10′. din 7.3

3. Să se demonstreze că mulțimea B a tuturor elementelor complementate în latricea distributivă $\langle P, 2, \vee, \wedge \rangle$ este închisă față de operațiile $\rightarrow$, $\vee$, $\wedge$, $\neg$ și că algebra $\langle B, 2, \vee, \wedge, \neg \rangle$ este o algebră booleană (v. 7.3, 7.2).

4. Să se demonstreze relațiile a), b), c) din demonstrația echivalenței definițiilor algebrelor $\mathscr{P}$ și $\mathscr{P}^*$ (v. 7.3).

5. Să se deducă p1 — p8 din p1* — p8* (v. 7.3).

6. Să se demonstreze $p'_1 — p'_{10}$ pentru o algebră Post de ordinul m (v. 7.3).

7. Să se refacă demonstrația echivalenței între algebrele Post $\mathscr{P}$ și $\mathscr{P}^*$ de ordinul m (v. 7.3).

8. Să se studieze proprietățile unei algebre Post de ordinul 4, analog cu metoda urmată pentru cea de ordinul 3 (v. 7.3).

9. Să se demonstreze că în orice algebră Post $\mathscr{P}$ pentru orice $a \in \mathscr{P}$ și pentru orice $j = 1, \ldots, n-1$ condiția $e_{j+1} \leqslant D_j(a) \vee e_i$ pentru $i \in \{0, \ldots, n-2\}$ implică $D_j(a) = e_{n-1}$ (v. 7.3).

Scanned with OKEN Scanner

10. Să se demonstreze relațiile a) — e), care sînt variante ale principiului contradicției (v. 7.4).

11. Să se demonstreze relațiile a') — e'), variante ale principiului terțului exclus (v. 7.4).

12. Să se deducă 110 din algebra LM de ordinul trei, pornind de la o algebră Post de ordinul 3 în care $\neg a = \neg D_2(a)c_1 \vee \neg D_1(a)c_2$ (v. 7.4).

13. Să se verifice axiomele $p1^*-p8^*$ ale algebrei Post de ordinul 3 pornind de la o algebră Łukasiewicz-Moisil de ordinul 3, unde

$$D_i(c_j) = \begin{cases} 2, & i \leqslant j,\ i = 1, 2, \\ 0, & i > j,\ j = 0, 1, 2, \text{ iar } C(a) = \neg D_1(a) \quad \text{(v. 7.4)}. \end{cases}$$

BIBLIOGRAFIE

1. Balbes, R., Dwinger, P. *Distributive Lattices*. Columbia, University of Missouri Press, 1974
2. Boicescu, V., Iorgulescu, A. *Cercetări actuale în domeniul algebrelor Lukasiewicz-Moisil* În : Probleme de logică, Vol. IX. București, Editura Academiei R.S.R., 1936.
3. Cignoli, R. *Moisil Algebras*. În : Notas de Logica Matematica Institute de Matematica, Universidad del sur Bahia Blanca, 27, 1970.
4. Epstein, G. *The Lattice Theory of Post Algebras*. În : Trans. Am. Math. Soc., Vol. 95, nr. 2, May, 1960, p. 300—317.
5. Iorgulescu, A. *On the construction of Three-Valued Lukasiewicz-Moisil Algebras*. În : Discrete Mathematics, 48, 1984, p. 213—227.
6. Moisil, Gr. C. *Algebre Lukasiewicziene*. În : Încercări vechi și noi de logică neclasică. București, Editura Științifică, 1965.
7. Monteiro, A. *Sur la définition des algèbres de Lukasiewicz trivalentes*. În : Bull. Acad. Pol. Sci. Sér. Sci. Math. Astr. Phys., 21, 1964.
8. Monteiro, L. *Axiomes indépendants pour des algèbres de Lukasiewicz trivalents*. Notas de Logica Matematica Instituto de Matematica, Universidad del sur Bahia Blanca, 22, 1964.
9. Prior, A. N. *Formal Logic*. Oxford at Clarendon Press, 1962.
10. Rosser, J. B., Turquette, A. R. *Many-Valued Logics*. North Holland, 1952.
11. Rousseau, G. *Post Algebras and Pseudo-Post Algebras*. În : Found. Math., 67, 1970, p. 133—145.
12. Rescher, N. *Many-Valued Logics*. New York, McGraw Hill, 1969.
13. Sicol, C. O. *Sur la définition des algèbres Lukasiewicziennes polivalentes*. În : Rev. Roumaine Math. Purves. Appl., 13, 1968, p. 1027—1030.
14. Traczyk, T. *Post Algebras trough P_0 and P_1 Lattices*. În : Computer Science and Multiple Valued Logic, Rine, D. (ed.), North Holland, 1977.

Scanned with OKEN Scanner

8.

PRIVIRE ASUPRA LOGICILOR MODALE

8.1. OPERATORI MODALI

În calculul propoziţiilor clasic utilizăm frecvent expresiile „p este tautologie“ echivalent cu „p este necesarmente adevărată“ sau „p este peste tot realizabilă”. Această afirmaţie este metalogică pentru că se referă din afara sistemului la proprietăţi ale propoziţiilor din sistem. Dacă notăm cu $\Box p$ noua propoziţie „p este peste tot realizabilă” (în toate interpretările), putem să concepem pe $\Box$ ca un nou operator logic monar, care aplicat unei propoziţii p generează o nouă propoziţie $\Box p$ şi să-l introducem ca un nou simbol în alfabetul CP.

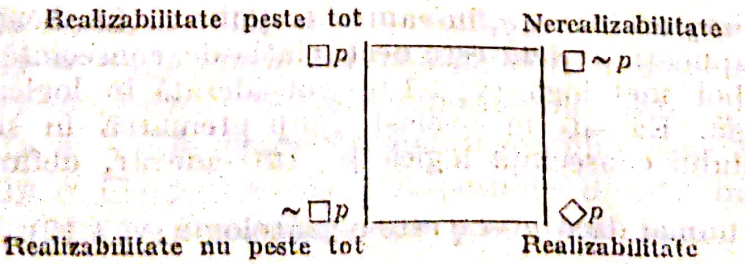
Cu această notaţie, contradicţia sau nerealizabilitatea se va enunţa: $\Box \sim p$ sau „$\sim p$ este peste tot realizabilă“ sau „p nu este nicăieri realizabilă“. Proprietatea de realizabilitate se va nota cu un nou simbol logic $\Diamond p$, citit „p este realizabilă” (în unele locuri sau interpretări). Realizabilitatea este negaţia nerealizabilităţii, deci avem următoarea relaţie fundamentală:

$$\Diamond p = \sim \Box \sim p$$

Propoziţia „nu este adevărat că p este peste tot realizabilă“ se scrie $\sim \Box p$ şi se mai poate citi „p este realizabilă nu peste tot“ sau „p este uneori realizabilă“.

Cei patru operatori $\Box p$, $\Box \sim p$, $\Diamond p$, $\sim \Box p$ se numesc *operatori modali*. Ei se referă la „măsura“ în care p este adevărată, de la latinescul „modus“ (măsură). Dar, construind modalitatea prin conceptul de realizabilitate, sensul ce se degajează nu este acela al unei măsuri mai mari sau mai mici de adevăr (sau de fals), ci acela al locurilor (interpretărilor) în care o propoziţie este adevărată (sau falsă). Modalitatea poate răspunde la întrebarea „cît de adevărat?“ sau „unde sau cînd este adevărat?“.

Operatorii modali sînt introduşi prin definiţia unuia dintre ei. Dacă introducem realizabilitatea $\Diamond p$, ceilalţi trei vor rezulta conform relaţiilor din următoarea versiune a *pătratului lui Boetius*:



În logica modală, denumirile operatorilor modali sînt :

necesarul	$\Box p$	νp	Lp
imposibilul	$\Box \sim p = \sim \Diamond p$	ηp	Up
contigentul	$\sim \Box p$	γp	Zp
posibilul	$\Diamond p = \sim \Box \sim p$	μp	Mp

Notaţiile alternative ale operatorilor modali au fost scrise în dreptul fiecăruia.

Operatorii modali, deşi formează noi formule atunci cînd sînt aplicaţi unor formule, nu au proprietatea de a fi funcţionale de adevăr, proprietate de care se bucură toţi operatorii binari cu care lucrăm în CP. Într-adevăr, adevărul sau falsul unei noi formule create de un operator binar era determinat în mod unic de adevărul sau falsul formulelor asupra cărora s-a operat, în conformitate cu tabla de adevăr. Nu putem întocmi tabla de adevăr pentru operatorii modali, din următorul motiv :

Cînd p este falsă, $\Box p$ este falsă, pentru că $\Box p$ afirmă că p este adevărată peste tot. Dar cînd p este adevărată, nu putem deduce că $\Box p$ este adevărată, căci din faptul că p este adevărată (într-o interpretare) nu rezultă că este adevărată peste tot.

Analog, cînd p este adevărată, $\Diamond p$ este adevărată, pentru că este suficient ca p să fie adevărată într-o singură interpretare, ca propoziţia „p este realizabilă“, adică $\Diamond p$ să fie adevărată. Dar dacă p este falsă, nu putem deduce că $\Diamond p$ este falsă, pentru că p poate fi adevărată într-o altă interpretare şi $\Diamond p$ să fie adevărată. Situaţia se rezumă astfel :

p	$\Box p$	$\Diamond p$		$\Box p$	p	şi	$\Diamond p$	p
1	?	1		1	1		1	?
0	0	?		0	?		0	0

De aici putem deduce că

$$\Box p \to p \text{ şi } p \to \Diamond p$$

pentru că p nu poate fi fals cînd $\Box p$ este adevărat şi $\Diamond p$ nu poate fi fals cînd p este adevărat. De altfel este suficient să avem una din relaţii pentru ca să o deducem pe cealaltă :

$$\text{din } \Box p \to p \text{ rezultă } \sim p \to \sim \Box p \text{ sau}$$

$$p \to \sim \Box \sim p \text{ şi deci } p \to \Diamond p.$$

Implicaţie modală. În logica modală intervine încă un simbol nou, care se adaugă la alfabetul CP, anume *implicaţie modală* sau *implicaţie strictă*, notată cu $\prec$. Prin definiţie

$$p \prec q \equiv \Box(p \to q)$$

A spune că p implică strict q, înseamnă a spune că implicaţia $p \to q$ este tautologie. Implicaţia strictă este deci relaţia de consecinţă $\Rightarrow$, care în CP era un simbol metalogic (v. 1.12), considerată în logica modală drept operator logic. Ea este în acelaşi timp preluarea în sistem ca operator al simbolului consecinţă logică $\models$. Într-adevăr, definiţia acestuia era

$p \models q$ dacă şi numai dacă $p \to q$ este o tautologie (v. 1.12).

Scanned with OKEN Scanner

Rolul relației $\Rightarrow$ în constituirea unei latice propoziționale a fost discutat în cap. 3.

Vom putea scrie în continuare $p \to q$ în loc de $p \prec q$, ori de cîte ori rezultă clar că $p \to q$ este o tautologie sau în sistemul formal o teză (axiomă sau teoremă).

O altă formă a definiției implicației stricte este

$$p \prec q \equiv \sim \Diamond (p \wedge \sim q)$$

(se citește „p implică strict q" este echivalent cu „nu este posibil ca p să fie adevărat și q fals"). Ea se deduce din prima definiție astfel :

$$p \prec q = \Box(p \to q) = \Box(\sim p \vee q) = \sim \Diamond \sim (\sim p \vee q) = \sim \Diamond (p \wedge \sim q).$$

Reducerea operatorilor modali. Orice șir de operatori modali formează tot un operator modal : $\Box\Diamond$, $\Box\Box\Diamond\Box\Diamond$ sau $(\sim\Box)\Diamond(\sim\Box)\Diamond\Box$ etc. Oprim proliferarea operatorilor modali introducînd cerința naturală $\Box\Box p \equiv \Box p$, întrucît este de așteptat ca „este necesar ca p să fie necesar" să fie echivalent cu „p este necesar".

Din $\Box\Box p \equiv \Box p$ și din definiția $\Box p \equiv \sim\Diamond \sim p$ deducem $\Diamond\Diamond p \equiv \Diamond p$, scriind succesiv

$$\sim\Box\Box\sim p \equiv \sim\Box\sim p$$
$$\Diamond\sim\Box\sim p \equiv \Diamond p$$
$$\Diamond\Diamond p \equiv \Diamond p$$

Introducem acum *principiul dublei imposibilități* :

$$\sim\Diamond\sim\Diamond p \equiv \Diamond p$$

care poate fi citit „imposibilitatea imposibilității este posibilitate".

Cu ajutorul formulelor de mai înainte stabilim alte relații de reducere a combinațiilor între operatorii modali. Astfel :

$$\Box\Diamond p = \sim\Box\sim\Diamond p = \Diamond p$$
$$\Diamond\Box p = \Diamond\sim\Diamond\sim p = \sim\Diamond\sim p = \Box p$$

Vom scrie acum regulile de reducere (cele stabilite anterior și altele care se deduc într-un mod analog) împreună cu interpretarea modală :

$\Box\Box p \equiv \Box p$	necesarul necesarului este necesar ;
$\Diamond\Diamond p \equiv \Diamond p$	posibilul posibilului este posibil ;
$\Box\Diamond p \equiv \Diamond p$	necesarul posibilului este posibil ;
$\Diamond\Box p \equiv \Box p$	posibilul necesarului este necesar ;
$\Box\sim\Box\sim p \equiv \Diamond p$	imposibilul imposibilului este posibil ;
$\sim\Box\sim\Box p \equiv \Box p$	contingentul contingentului este necesar ;
$\sim\Box\Box\sim p \equiv \Diamond p$	contingentul imposibilului este posibil ;
$\Box\sim\sim\Box p \equiv \Box p$	imposibilul contingentului este necesar ;
$\Diamond\sim\Box p \equiv \sim\Box p$	posibilul contingentului este contigent ;
$\Diamond\Box\sim p \equiv \Box\sim p$	posibilul imposibilului este imposibil ;
$\Box\sim\Box p \equiv \sim\Box p$	necesarul contingentului este contingent ;
$\Box\Box p \equiv \Box\sim p$	necesarul imposibilului este imposibil ;
$\sim\Box\Diamond p \equiv \Box\sim p$	contingentul posibilului este imposibil ;

Scanned with OKEN Scanner

$\sim\Box\Box p \equiv \sim\Box p$ contingentul necesarului este contingent;
$\Box\sim\Diamond p \equiv \Box\sim p$ imposibilul posibilului este imposibil;
$\Box\sim\Box p \equiv \sim\Box p$ imposibilul necesarului este contingent.

Relațiile între operatori modali și operatori clasici. Vom cere ca operatorul „necesitate" să fie distributiv față de conjuncție:

$$\Box(p \wedge q) = \Box p \wedge \Box q.$$

De aici va rezulta distributivitatea lui $\Diamond$ față de disjuncție:

$$\Diamond(p \vee q) = \Diamond p \vee \Diamond q.$$

Să deducem relația a doua din prima:

$$\Box(\sim p \wedge \sim q) = \Box\sim p \wedge \Box\sim q$$

$$\Box\sim(p \vee q) = \Box\sim p \wedge \Box\sim q$$

$$\sim\Diamond(p \vee q) = \sim\Diamond p \wedge \sim\Diamond q$$

Deci

$$\Diamond(p \vee q) = \Diamond p \vee \Diamond q.$$

Legătura cu implicația este dată astfel:

dacă $p \to q$, atunci $\Box p \to \Box q$.

De aici va rezulta

dacă $p \to q$, atunci $\Diamond p \to \Diamond q$.

Într-adevăr, dacă $p \to q$ și $\Box p \to \Box q$, atunci $\sim q \to \sim p$ și $\Box\sim q \to \Box\sim p$, ca urmare $\sim\Box\sim p \to \sim\Box\sim q$, ceea ce este conform definiției lui $\Diamond$: $\Diamond p \to \Diamond q$.

Conform convenției:

$$\Box p \to (p \to \Diamond p)$$

și deci $\Box p \to \Diamond p$ (citim „necesarul implică posibilul").

De aici rezultă că „imposibilul implică contingentul":

$$\Box\sim p \to \Diamond\sim p = \sim\Box p.$$

8.2. AXIOME ȘI REGULI

Construim un sistem logic formal cu modalitate bazîndu-ne pe axiomele unui sistem formal al logicii propozițiilor, la care adăugăm un număr minimal de axiome, reglementînd comportamentul operatorului modal introdus. Alegem ca axiome ale sistemului logicii propoziționale fără modalitate axiomele din *Principia Mathematica* (v. 2.3), avînd același alfabet de simboluri primitive:

CP1 $(p \vee p) \to p$

CP2 $q \to (p \vee q)$

CP3 $(p \vee q) \to (q \vee p)$

CP4 $(q \to r) \to ((p \vee q) \to (p \vee r))$

Scanned with OKEN Scanner

$\sim\Box\Box p \equiv \sim\Box p$ contingentul necesarului este contingent;
$\Box\sim\Diamond p \equiv \Box\sim p$ imposibilul posibilului este imposibil;
$\Box\sim\Box p \equiv \sim\Box p$ imposibilul necesarului este contingent.

Relațiile între operatori modali și operatori clasici. Vom cere ca operatorul „necesitate" să fie distributiv față de conjuncție:

$$\Box(p \wedge q) = \Box p \wedge \Box q.$$

De aici va rezulta distributivitatea lui $\Diamond$ față de disjuncție:

$$\Diamond(p \vee q) = \Diamond p \vee \Diamond q.$$

Să deducem relația a doua din prima:

$$\Box(\sim p \wedge \sim q) = \Box\sim p \wedge \Box\sim q$$
$$\Box\sim(p \vee q) = \Box\sim p \wedge \Box\sim q$$
$$\sim\Diamond(p \vee q) = \sim\Diamond p \wedge \sim\Diamond q$$

Deci

$$\Diamond(p \vee q) = \Diamond p \vee \Diamond q.$$

Legătura cu implicația este dată astfel:

dacă $p \to q$, atunci $\Box p \to \Box q$.

De aici va rezulta

dacă $p \to q$, atunci $\Diamond p \to \Diamond q$.

Într-adevăr, dacă $p \to q$ și $\Box p \to \Box q$, atunci $\sim q \to \sim p$ și $\Box\sim q \to \Box\sim p$, ca urmare $\sim\Box\sim p \to \sim\Box\sim q$, ceea ce este conform definiției lui $\Diamond$: $\Diamond p \to \Diamond q$.

Conform convenției:

$$\Box p \to (p \to \Diamond p)$$

și deci $\Box p \to \Diamond p$ (citim „necesarul implică posibilul").

De aici rezultă că „imposibilul implică contingentul":

$$\Box\sim p \to \Diamond\sim p = \sim\Box p.$$

8.2. AXIOME ȘI REGULI

Construim un sistem logic formal cu modalitate bazîndu-ne pe axiomele unui sistem formal al logicii propozițiilor, la care adăugăm un număr minimal de axiome, reglementînd comportamentul operatorului modal introdus. Alegem ca axiome ale sistemului logicii propoziționale fără modalitate axiomele din *Principia Mathematica* (v. 2.3), avînd același alfabet de simboluri primitive:

CP1 $(p \vee p) \to p$

CP2 $q \to (p \vee q)$

CP3 $(p \vee q) \to (q \vee p)$

CP4 $(q \to r) \to ((p \vee q) \to (p \vee r))$

Scanned with OKEN Scanner

Modalitatea (necesitatea) este introdusă prin axiomele

A0 $\quad \Box(p \to q) \to (\Box p \to \Box q)$

A2 $\quad \Box p \to p$ (axioma necesității)

Axioma A0 nu este decît o altă formă a lui

$$\Box p \wedge \Box(p \to q) \to \Box q$$

care este citirea modală a lui modus ponens :

$$\text{dacă } p \text{ și } p \to q, \text{ atunci } q.$$

Sistemul modal definit de axiomele CP1 — CP4, A0 — A2 se numește *sistemul* **T.**

Regulile de transformare în **T** sînt regulile clasice : R1, R2, modus ponens, la care se adaugă regula

R3 $\quad \vdash \alpha \Rightarrow \vdash \Box\alpha,$

citită astfel : dacă α este o teoremă, atunci și $\Box\alpha$ este o teoremă. În mod practic vom recurge și la regula dedusă

R4 $\quad \vdash(\alpha \to \beta) \to \vdash(\Box\alpha \to \Box\beta).$

Derivarea lui R4 este imediată :

dacă $\vdash(\alpha \to \beta)$, atunci $\vdash \Box(\alpha \to \beta)$, (R3)

dar $\Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta)$ (A6)

și deci $\vdash(\Box\alpha \to \Box\beta).$ (R2)

Dacă la axiomele lui **T** adăugăm axioma

A3 $\quad \Box p \to \Box\Box p,$

sistemul definit de A1 — A7 și aceleași reguli de bază R1 — R3 se numește **S4.**

Dacă la axiomele lui **T** adăugăm axioma

A5 $\quad \Diamond p \to \Box\Diamond p,$

sistemul definit de A1 — A6 plus A8 poartă denumirea de **S5.**

Modalitatea $\Diamond$, implicația strictă și egalitatea modală sînt introduse prin definiție :

$$\Diamond\alpha = \sim\Box\sim\alpha$$

$$\alpha \prec \beta = \Box(\alpha \to \beta)$$

$$(\alpha \equiv \beta) = (\alpha \prec \beta) \wedge (\beta \prec \alpha)$$

Întrucît axiomele sistemelor **T**, **S4** și **S5** conțin axiomele logicii propoziționale clasice, orice teoremă a acesteia va fi și teoremă a celor trei sisteme. Întrucît axiomele lui **S4** și **S5** conțin axiomele lui **T**, orice teoremă a lui **T** este teoremă și în **S4** și **S5**. Spunem că **T** este un *sistem slab*, iar **S4** și **S5** *sisteme tari*. Nu orice teoremă a lui **S5** este și teoremă în **S4**, dar orice teoremă a lui **S4** este teoremă în **S5**, întrucît axioma A3 este teoremă în **S5**.

Scanned with OKEN Scanner

8.3. SISTEMUL T

Vom da o listă de teoreme deduse din axiomele sistemului **T** cu ajutorul regulilor de transformare. Fiind vorba numai de teze (axiome, teoreme), vom putea utiliza în scriere implicația materială. Demonstrația va fi dată ilustrativ la un număr restrins de teoreme :

1. $p \to \Diamond p$

2. $(p \equiv q) \to (\Box p \equiv \Box q)$

3. $\Box(p \wedge q) \equiv \Box p \wedge \Box q$

Demonstrație.

a) $\Box(p \wedge q) \to \Box p$ — din CP : $p \wedge q \to p$ și R4

b) $\Box(p \wedge q) \to \Box q$ — idem, rezultă :

c) $\Box(p \wedge q) \to \Box p \wedge \Box q$

d) $\Box p \to \Box(q \to (p \wedge q))$ — din CP și R4

e) $\Box(q \to (p \wedge q)) \to (\Box q \to \Box(p \wedge q))$ — A0, substituție

f) $\Box p \to (\Box q \to \Box(p \wedge q))$ — silogismul ipotetic (v. 2.5)

g) $\Box p \wedge \Box q \to \Box(p \wedge q)$ — din f)

h) $\Box(p \wedge q) \equiv \Box p \wedge \Box q$ — din c) și g)

4. $\Box(p \equiv q) \equiv (p \equiv q)$

5. $\Box p \equiv \sim \Diamond \sim p$

6. $\Box \sim p \equiv \sim \Diamond \sim p$

7. $\sim \Box p \equiv \Diamond \sim p$

8. $\Box \Box p \equiv \sim \Diamond \Diamond \sim p$

9. $\Box \Box \sim p \equiv \sim \Diamond \Diamond p$

10. $\Diamond \Diamond \sim p \equiv \sim \Diamond \Diamond p$

11. $\Box \Diamond \sim p \equiv \sim \Diamond \Box p$

12. $\Diamond \Box \sim p \equiv \sim \Box p$

13. $\sim \Diamond(p \vee q) \equiv \sim \Diamond p \wedge \sim \Diamond q$

Demonstrație.

a) $\Box(\sim p \wedge \sim q) \equiv \Box p \wedge \Box \sim q$ — teorema 3, substituție

b) $\sim \Diamond \sim (\sim p \wedge \sim q) \equiv (\sim \Diamond p \wedge \sim \Diamond q)$ — definiție

c) $\sim \Diamond(p \vee q) \equiv \sim \Diamond p \wedge \sim \Diamond q$ — CP

14. $\Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$

Demonstrație.

a) $(\sim p \equiv q) \to (p \equiv \sim q)$ — CP

b) $\Diamond(p \vee q) \equiv \sim(\sim \Diamond p \wedge \sim \Diamond q)$ — din a) și teorema 13

c) $\Diamond(p \vee q) = \Diamond p \vee \Diamond q$ — CP

Scanned with OKEN Scanner

... $p \prec q \rightarrow (\Diamond p \rightarrow \Diamond q)$

16. $\Box p \vee \Box q \rightarrow \Box(p \vee q)$.

Demonstrație.

a) $p \rightarrow p \vee q$ — CP

b) $\Box p \rightarrow \Box(p \vee q)$ — R4

c) $q \rightarrow p \vee q$ — CP2

d) $\Box q \rightarrow \Box(p \vee q)$ — R4

e) $(p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$ — CP

f) $\Box p \vee \Box q \rightarrow \Box(p \vee q)$ — din b), d) și e)

17. $\Diamond(p \wedge q) \rightarrow \Diamond p \wedge \Diamond q$

Demonstrație.

a) $\Box \sim p \vee \Box \sim q \rightarrow \Box(\sim p \vee \sim q)$ — teorema 16, substituție

b) $\sim \Box(\sim p \vee \sim q) \rightarrow \sim(\Box \sim p \vee \Box \sim q)$ — din a)

c) $\Diamond \sim(\sim p \vee \sim q) \rightarrow \sim(\Diamond p \vee \sim \Diamond \sim q)$ — definiție

d) $\Diamond(p \wedge q) \rightarrow \Diamond p \wedge \Diamond q$.

18. $(\sim p \prec p) \equiv \Box p$

19. $p \prec \sim p \equiv \Box \sim p$

20. $(q \prec p) \wedge (\sim q \prec p) \equiv \Box p$

21. $(p \prec q) \wedge (p \prec \sim q) \equiv \Box \sim p$

22. $\Box p \rightarrow (q \prec p)$

23. $\Box \sim p \rightarrow (p \prec q)$

24. $\Box p \rightarrow (\Diamond q \rightarrow \Diamond(p \wedge q))$.

Tot în cadrul sistemului **T**, din teoremele stabilite se derivează reguli suplimentare de transformare a formulelor modale:

$\vdash(\alpha \equiv \beta) \rightarrow \vdash(\Box \alpha \equiv \Box \beta)$ — din teorema 4

$\vdash(\alpha \rightarrow \beta) \rightarrow \vdash(\Diamond \alpha \rightarrow \Diamond \beta)$ — din teorema 15

$\vdash \alpha \rightarrow \vdash(\Diamond \beta \rightarrow \Diamond(\alpha \wedge \beta))$ — din teorema 24

Inconvenientul principal al sistemului **T** constă în faptul că admite o infinitate de modalități: orice concatenare de simboluri modale este un nou operator modal (de exemplu $\Diamond\Box\Box\Diamond\Box\Diamond\ \Diamond\Diamond\Box p$). Relațiile care se deduc ușor din axioma $\Box p \rightarrow p$ și din teorema $p \rightarrow \Diamond p$ sînt:

25. $\Box\Diamond p \rightarrow \Diamond p$

26. $\Box p \rightarrow \Diamond\Box p$

27. $\Diamond p \rightarrow \Diamond\Diamond p$

28. $\Box\Box p \rightarrow \Box p$

Scanned with OKEN Scanner

8.4. SISTEMELE S4 ȘI S5

Introducînd axioma specifică a sistemului **S4**:

A3 $\quad \Box p \to \Box\Box p,$

deducem din A3 și 28 (v. 8.3):

1. $\quad \Box\Box p \equiv \Box p$

Se demonstrează imediat

2. $\quad \Diamond\Diamond p \equiv \Diamond p$

Demonstrație.

a) $\Box\sim p \to \Box\Box\sim p$ — A3, substituție

b) $\sim\Diamond p \to \sim\Diamond\Diamond p$ — definiție

c) $\Diamond\Diamond p \to \Diamond p$ — d in b

d) $\Diamond p \to \Diamond\Diamond p$ — 27

e) $\Diamond\Diamond p \equiv \Diamond p$

Următoarele teoreme permit noi reduceri:

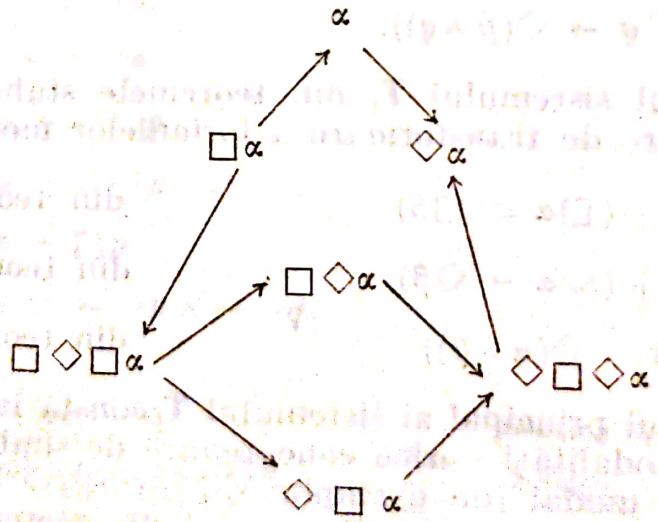
3. $\Diamond\Box\Diamond p \to \Diamond p$

4. $\Box\Diamond p \to \Box\Diamond\Box\Diamond p$

5. $\Box\Diamond p \equiv \Box\Diamond\Box\Diamond p$

6. $\Diamond\Box p \equiv \Diamond\Box\Diamond\Box p$

Continuă să fie operatori modali ireductibili alături de $\Box$ și $\Diamond$: $\Box\Diamond$, $\Diamond\Box$, $\Box\Diamond\Box$, $\Diamond\Box\Diamond$. Următoarea schemă arată implicațiile ce se stabilesc între operatorii pozitivi:



Sistemul S5. Axioma A5 caracteristică lui **S5** este mai tare decît A3, în sensul că odată introdusă între axiome, A3 poate fi dedusă ca teoremă, reciproca nefiind adevărată. Efectele lui A5 sînt următoarele:

A8′ $\quad \Diamond p \to \Box\Diamond p$

1. $\Box\Diamond p \equiv \Diamond p$ se obține din A5 și 25 (**S4**).

2. $\Diamond\Box p \equiv \Box p$

Scanned with OKEN Scanner

3. $\Box\Box p \equiv \Box p$

4. $\Diamond\Diamond p \equiv \Diamond p$

În **S5** vom avea deci două simboluri modale : $\Box$ și $\Diamond$.

În **T, S4** și **S5** sînt stabilite următoarele teoreme privind distributivitatea acestor operatori :

$\Box(p \wedge q) \equiv \Box p \wedge \Box q$ — **3T**

$\Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$ — **14T**

$\Box p \vee \Box q \rightarrow \Box(p \vee q)$ — **16T**

$\Diamond(p \wedge q) \rightarrow \Diamond p \wedge \Diamond q$ — **17T**

În **S5** se pot demonstra noi teoreme utile în calcul :

5. $\Box(p \vee q) \rightarrow \Box p \vee \Diamond q$

Demonstrație

a) $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$ — A0

b) $\Box(\sim q \rightarrow p) \rightarrow (\Box \sim q \rightarrow \Box p)$ — substituție

c) $\Box(q \vee p) \rightarrow (\sim\Box\sim q \vee \Box p)$ — definiție

d) $\Box(p \vee q) \rightarrow (\Box p \vee \Diamond q)$ — din c), definiție

6. $\Box(p \vee \Box q) \equiv \Box p \vee \Box q$

Demonstrație.

a) $\Box(p \vee \Box q) \rightarrow \Box p \vee \Box q$ — din 5, substituție și absorbție

b) $\Box p \vee \Box\Box q \rightarrow \Box(p \vee \Box q)$ — din 16T și substituție

c) $\Box p \vee \Box q \rightarrow \Box(p \vee \Box q)$ — absorbție $\Box\Box = \Box$

d) $\Box(p \vee \Box q) \equiv \Box p \vee \Box q$ — din a) și c).

7. $\Box(p \vee \Diamond q) \equiv \Box p \vee \Diamond q$

Demonstrație

a) $\Box(p \vee \Box\Diamond q) \equiv \Box p \vee \Box\Diamond q$ — din 6 și substituție

b) $\Box(p \vee \Diamond q) \equiv \Box p \vee \Diamond q$ — absorbție

8. $\Diamond(p \wedge \Diamond q) = \Diamond p \wedge \Diamond q$

Demonstrație

a) $\Box(\sim p \vee \Box \sim q) \equiv \Box \sim p \vee \Box \sim q$ — din 6 și substituție

b) $\sim\Box(\sim p \vee \Box \sim q) \equiv \sim(\Box \sim p \vee \Box \sim q)$

c) $\Diamond\sim(\sim p \vee \sim\Diamond q) \equiv \sim(\sim\Diamond p \vee \sim\Diamond q)$ — definiție

d) $\Diamond(p \wedge \Diamond q) \equiv \Diamond p \wedge \Diamond q$ — legile lui De Morgan

9. $\Diamond(p \wedge \Box q) \equiv \Diamond p \wedge \Box q$

Demonstrație

a) $\Diamond(p \wedge \Diamond\Box q) \equiv \Diamond p \wedge \Diamond\Box q$ — din 8 și substituție

b) $\Diamond(p \wedge \Box q) \equiv \Diamond p \wedge \Box q$ — absorbție

Scanned with OKEN Scanner

8.5. ALTE SISTEME MODALE

Sistemele modale cele mai cunoscute **T**, **S4**, **S5** s-au format din axiomele CP suplimentate cu axiome specifice modale și reguli de inferență. În cele ce urmează sînt prezentate o listă de axiome și reguli de inferență care stau la baza formării altor sisteme modale și apoi un tabel al acestora (tabelul 8.1), împreună cu denumirile sistemelor generate.

Axiome

A0 $\Box(p \to q) \to (\Box p \to \Box q)$

A1 $\Box p \to \Diamond p$

A2 $\Box p \to p$

A3 $\Box p \to \Box\Box p$

A4 $p \to \Box\Diamond p$

A5 $\Diamond p \to \Box\Diamond p$

A6 $\Box(\Box p \to p)$ sau A2

A7 $\Box p \to \Box(\Box q \to \Box p)$

A8 $\Box(\Box p \to \Box(\Box q \to \Box p))$ sau A7

A9 $(\Box(p \vee q) \wedge \Box(p \vee \Box q) \wedge \Box(\Box p \vee q)) \to (\Box p \vee \Box q)$

A10 $\Box(\Box(p \to \Box p) \to p) \to \Box(\Diamond\Box p \to p)$

Reguli

$$\text{R1}: \frac{A, A \to B}{B}, \quad \text{R2}: \frac{A \to B}{\Box A \to \Box B}, \quad \text{R3}: \frac{A}{\Box A}$$

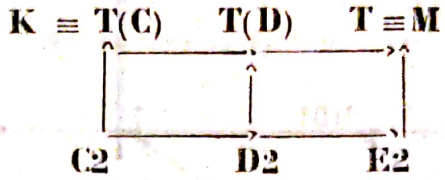
$$\text{R2}': \frac{\Box(A \to B)}{\Box(\Box A \to \Box B)}$$

Tabelul 8.1

Sisteme modale	Axiome CP1-4 +	Reguli
C2	A0	R_1, R_2
D2	A0, A1	R_1, R_2
E2	A0, A2	R_1, R_2
K ≡ T(C)	A0	R_1, R_3
T(D)	A0, A1	R_1, R_3
T ≡ M	A0, A2	R_1, R_3
S2	A0, A2	R_1, R_2'
S3	A0, A2, A6, A8	R_1
S4	A0, A2, A3	R_1, R_3
B	A0, A2, A4	R_1, R_3
S5	A0, A2, A5	R_1, R_3
S4.3	A0, A2, A3, A9	R_1, R_3
D	A0, A2, A3, A9, A10	R_1, R_3
S0.5	A0, A2	R_3

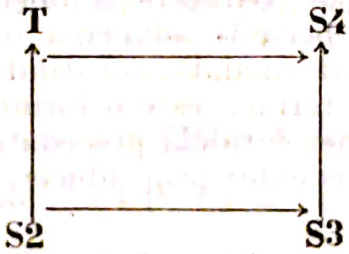
Scanned with OKEN Scanner

Comentariu. Primele șase sisteme prezintă o simetrie : primele trei **C2**, **D2**, **E2** au aceleași axiome ca următoarele trei **T(C)**, **T(D)**, **T**, dar au regulile R1 și R2, pe cînd ultimele au regulile R1 și R3. Întrucît R2 se obține din R1, R3 și A0 și A1 este o axiomă mai slabă decît A2, obținem următoarea schemă :

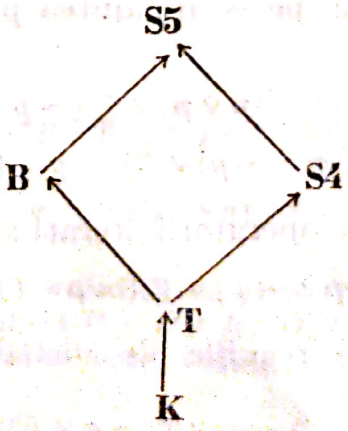


unde săgeata indică incluziunea unui sistem mai slab într-unul mai tare. (De exemplu **C2** → **K** înseamnă că toate teoremele lui **C2** sînt teoreme și în **K**.) În **C2** și **T(C)** axioma A1 nu este presupusă, în **D2** și **T(D)**, sisteme numite *deontice*, A1 este o axiomă, iar în **E2** și **T** ea este înlocuită cu A2 mai tare. Sistemul **T(C)** este numit în literatură **K** (de la Kripke), iar **T** poate fi întîlnit sub denumirea de **M** (sistemul Feys-von Wright).

Sistemele **S2**, **S3**, **S4**, **T** sînt pe baza definiției lor legate de următoarea schemă :



Din **T** prin adăugarea axiomei A3 se obține sistemul **S4** iar prin adăugarea axiomei A4 se obține sistemul **B** (sistemul Brouwer). Axioma A4 poartă numele de *axioma lui Brouwer*. Atît **B** cît și **S4** sînt cuprinse în **S5**. Legătura între aceste sisteme și sistemul **K** (inclus în toate) este reprezentată de schema



Sistemele **S4.3** și **D** sînt intermediare între **S4** și **S5**. Acestea au fost amintite pentru că apar în sistemele temporale. Sistemul **S.05** a fost inclus ca reprezentînd cel mai slab sistem cu axiomele A0, A2 și o singură regulă de inferență.

Scanned with OKEN Scanner

Schema din fig. 8.1 indică relațiile dintre toate sistemele introduse.

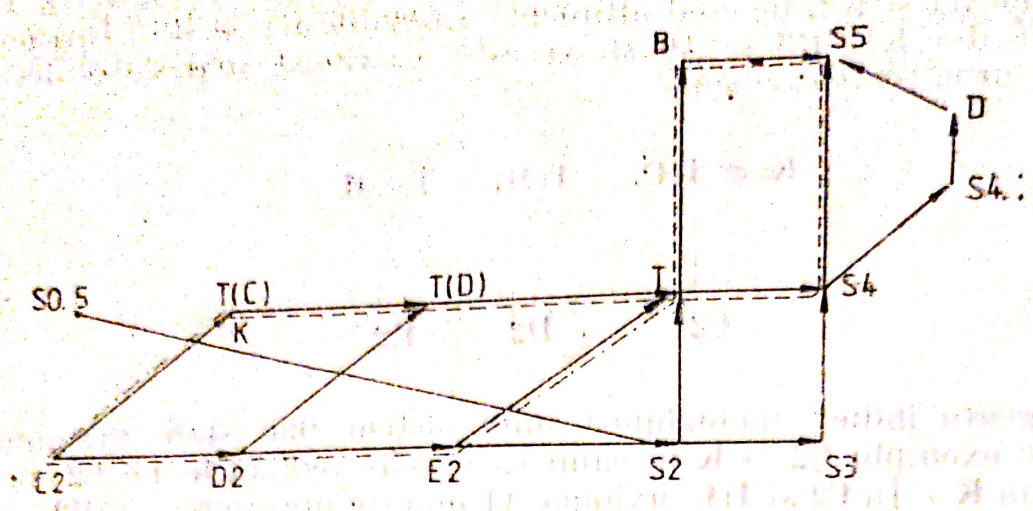


Fig. 8.1

8.6. FORMA NORMALĂ CONJUNCTIVĂ MODALĂ (FNCM)

Existența teoremelor de reducere a operatorilor modali compuși la operatorii simpli $\Box$, $\Diamond$ permite aducerea oricărei formule din **S5** la o formă normală conjunctivă modală, constînd într-un produs de clauze disjunctive, în care fiecare termen este o formulă din calculul propozițiilor nemodal sau o asemenea formulă precedată de $\Box$ sau $\Diamond$.

Vom ilustra pașii procedeului prin aducerea la forma FNCM a formulei

$$(\Diamond\Diamond p \prec p) \rightarrow (p \prec \Box p)$$

P a s u l 1. Eliminăm toți operatorii în afară de $\sim$, $\Box$, $\Diamond$, $\vee$, $\wedge$:

$$\Box(\Diamond\Diamond p \rightarrow p) \rightarrow \Box(p \rightarrow \Box p) \qquad \text{definiție}$$

$$\sim\Box(\sim\Diamond\Diamond p \vee p) \vee \Box(\sim p \vee \Box p) \qquad \text{definiție}$$

P a s u l 2. Eliminăm pe $\sim$ dinaintea parantezei sau a unui operator modal:

$$\Box\sim(\sim\Diamond\Diamond p \vee p) \vee \Box(\sim p \vee \Box p) \qquad \text{definiție}$$

$$\Diamond(\Diamond\Diamond p \wedge \sim p) \vee \Box(\sim p \vee \Box p) \qquad \text{De Morgan}$$

P a s u l 3. Reducem operatorii iterați:

$$\Diamond(\Diamond p \wedge \sim p) \vee \Box(\sim p \vee \Box p)$$

P a s u l 4. Aplicăm regulile de distribuție (inclusiv teoremele 6—9, **S5**):

$$(\Diamond p \wedge \Diamond\sim p) \vee \Box\sim p \vee \Box p$$

$$(\Diamond p \vee \Box\sim p \vee \Box p) \wedge (\Diamond\sim p \vee \Box\sim p \vee \Box p) \qquad \text{FNCM}$$

Dacă la pasul 4 formula mai are expresii de forma $\Box\alpha$ sau $\Diamond\alpha$, unde α este conjuncție sau disjuncție conținînd operatori modali, distin-

Scanned with OKEN Scanner

gem următoarele cazuri :

1) $\Box\alpha$ și α conjuncție ; distribuim ;

2) $\Box\alpha$ și α disjuncție cu un operator modal : distribuim conform teoremelor 6—9 (S5) ca în exemplul tratat ;

3) $\Box\alpha$ și α disjuncție, în care termenii nu încep cu un operator modal : aplicăm formula CP

$$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$$

și distribuim.

Exemplu

$$\Box(p \vee (q \wedge \Diamond r) = \Box(p \vee q) \wedge \Box(p \vee \Diamond r) = \Box(p \vee q) \wedge (\Box p \vee \Diamond r).$$

Procedeul de decizie bazat pe FNCM. Îl vom ilustra prin testul validității formulei :

$$M = (\Diamond p \vee \Diamond(p \wedge \sim q) \vee \Box(\sim p \vee q) \vee \Diamond r) \wedge (\Box \sim r \vee \Diamond(p \wedge \sim q) \vee \vee \Box(\sim p \vee q) \vee \Diamond r).$$

Pasul 1. Aducerea formulei la forma FNCM. Ea este în această formă. Notăm cele două clauze modale C_1, C_2: $M = C_1 \wedge C_2$.

Pasul 2. Ordonăm elementele fiecărei clauze, începînd cu cele care nu sînt sub incidența operatorilor modali și continuînd cu $\Box$ și apoi cu $\Diamond$:

$$C_i = P \vee \Box Q_1 \vee \Box Q_2 \vee \ldots \vee \Box Q_n \vee \Diamond R_1 \vee \Diamond R_2 \vee \ldots \vee \Diamond R_m,$$

unde P, Q_j, R_l sînt formule din CP. La acest pas

$$M = (\Box(\sim p \vee q) \vee \Diamond p \vee \Diamond(p \wedge \sim q) \vee \Diamond r)) \wedge (\Box \sim r \vee \Box(\sim p \vee q) \vee \vee \Box(p \wedge \sim q) \vee \Diamond r).$$

Pasul 3. Distribuim $\Diamond$ și clauzele devin

$$C_i = P \vee \Box Q_1 \vee \ldots \vee \Box Q_n \vee \Diamond R.$$

În cazul nostru :

$$M = (\Box(\sim p \vee q) \vee \Diamond(p \vee p \wedge \sim q \vee r)) \wedge (\Box \sim r \vee \Box(\sim p \vee q) \vee \vee \Diamond((p \wedge \sim q) \vee r).$$

Pasul 4. Examinăm validitatea fiecărei clauze astfel :

Formăm $n + 1$ formule CP : $P \vee R$, $Q_1 \vee R$, ..., $Q_n \vee R$. Dacă cel puțin una este validă, clauza este validă.

$$C_1 = \Box(\sim p \vee q) \vee \Diamond((p \vee p) \wedge (\sim q \vee r))$$

$$P = 0,\ Q_1 = \sim p \vee q,\ R = p \vee p \wedge \sim q \vee r$$

$Q_1 \vee R = \sim p \vee q \vee p \vee (p \wedge \sim q) \vee r = 1$, deci C_1 este validă.

$C_2 = \Box \sim r \vee \Box(\sim p \vee q) \vee \Diamond(p \wedge \sim q \vee r)$
$P = 0,\ Q_1 = \sim r,\ Q_2 = \sim p \vee q,\ R = p \wedge \sim q \vee r$
$Q_1 \vee R = \sim r \vee p \wedge \sim q \vee r = 1$, deci C_2 este validă.
$Q_2 \vee R = \sim p \vee q \vee \sim p \wedge q \vee r = 1$

Pasul 5. Dacă fiecare clauză modală este validă, formula este validă.

În cazul considerat $M = C_1 \wedge C_2$ este validă.

8.7. PROCEDURA DECIZIEI PRIN DIAGRAME

Procedura prin diagrame semantice este asemănătoare cu cea utilizată în calculul propozițiilor pentru demonstrarea caracterului de tautologie (sau contradicție) a unei formule (v.1.5). Metoda constă în adoptarea ipotezei inițiale că formula dată este o contradicție. Se atribuie în consecință valorile corespunzătoare subformulelor principale și apoi părților lor pînă cînd se ajunge la variabile. Dacă în acest fel se obține o contradicție, înseamnă că încercarea de falsificare a formulei nu a reușit și că formula este validă.

Să luăm ca exemplu formula

$$\Diamond p \wedge \Diamond \sim p \rightarrow \sim \Box p$$

Presupunem că este falsă. Vom scrie sub conectorii principali valorile ce se impun :

$$\underset{1}{\Diamond p} \wedge \underset{1}{\Diamond} \sim p \rightarrow \underset{0}{\sim} \underset{0}{\Box p}$$

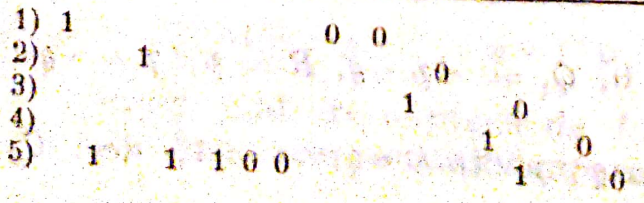
întrucît o implicație este falsă numai cînd membrul întîi este 1 și membrul doi este zero. Din $\sim \Box p$ rezultă că $\Box p$ este 1, ceea ce atrage după sine $p = 1$. Membrul întîi este 1 cînd ambele conjuncții sînt 1. A doua conjuncție este $\Diamond \sim p$, dar $p = 1$ și $\Diamond \sim p$ este $\Diamond 0 = 0$. Am ajuns astfel la o contradicție pe care o subliniem sub formulă :

$$\begin{array}{c} p \wedge \Diamond \sim p \rightarrow \sim \Box p \\ 1\ 0\ 0\ 1 \rightarrow\ 0\ 1\ 1 \end{array}$$

Observăm că subformulele $\underset{1}{\Box p}$ și $\underset{0}{\Diamond p}$ determină valoarea lui p, dar $\underset{0}{\Box p}$ și $\underset{1}{\Diamond p}$ nu determină valoarea lui p. În primul caz punem un asterisc deasupra operatorilor modali, iar în al doilea un asterisc dedesubtul lor. Asteriscurile de dedesubt ne trimit la o nouă variantă. Pentru a distinge variantele, formulele se încadrează în dreptunghiuri (blocuri) legate de săgeți pornite de la asteriscurile de jos.

Exemplu. Fie formula sistemului T

$$\Box(p \rightarrow \Diamond(q \rightarrow r)) \rightarrow \Diamond(q \rightarrow (\Box p \rightarrow \Diamond r))$$



Deşi valorile se scriu într-un singur rînd, sînt scoase în evidenţă cele 5 etaje în care acestea sînt atribuite conform ipotezei. Se plasează asteriscurile după regula enunţată :

$$\boxed{\begin{array}{l} \overset{*}{\Box}(p \to \Diamond(q \to r)) \to \Diamond(q \to (\Box p \to \Diamond r)) \\ 1\ 1\ 1\ \ 1\ 1\ 0\ 0\ \ 0\ 1\ 1\ 0\ \ 1\ 1\ 0\ 0\ 0 \end{array}}$$

Nu există nici o contradicţie pentru că $\Diamond(q \to r)$, cînd $q \to r$, este 0, poate fi şi 0 şi 1. De aceea coborîm o săgeată spre o nouă diagramă întocmită după regulile următoare :

1) În despărţiturile noii diagrame se scriu formulele de sub incidenţa operatorului ambiguu, urmate de formulele de sub incidenţa operatorilor cu asterisc sus (fără aceşti operatori).

2) Tuturor formulelor li se atribuie în noua diagramă valorile pe care le-a avut operatorul lor în diagrama anterioară :

În vechea interpretare	$\overset{*}{\underset{1}{\Box}}\alpha$	$\overset{*}{\underset{0}{\Diamond}}\alpha$	$\underset{0}{\Box}\alpha$	$\underset{1}{\Diamond}\alpha$
În noua interpretare	$\alpha = 1$	$\alpha = 0$	$\overset{*}{\alpha = 0}$,	$\overset{*}{\alpha = 1}$

În exemplu subformulele p (de sub $\overset{*}{\underset{1}{\Box}}p$) vor primi valoarea 1, r (de sub $\overset{*}{\underset{0}{\Diamond}}q$) va primi valoarea 0, $q \to (\Box p \to \Diamond r)$ (de sub $\overset{*}{\underset{0}{\Diamond}}(q \to \Box p \to \Box r)$) va primi valoarea 0, $p \to \Diamond(q \to r)$ (de sub $\overset{*}{\underset{1}{\Box}}(p \to \Diamond(q \to r))$) va primi valoarea 1 şi, în sfîrşit, lui $q \to r$ (de sub $\underset{1}{\Diamond}(q \to r)$) i se va atribui valoarea 1.

Diagrama va arăta astfel :

$q \to r$	$p \to \Diamond(q \to r)$	$q \to (\Box p \to \Diamond r)$	p	r
1 1 0	1	1 0	1	0

De data aceasta s-a obţinut o contradicţie şi deci formula iniţială este validă. Inexistenţa unei contradicţii în primul bloc nu a fost concludentă. Acest lucru corespunde definiţiei validităţii în logica modală, conform căreia validitatea depinde nu numai de valoarea de adevăr a unei propoziţii, ci şi de „lumile posibile" în care se face evaluarea, lumi reprezentate în cazul nostru de blocurile succesive (v. 8.4).

Esenţa metodei constă în a trece de la o interpretare în lumea posibilă w_1 la o altă interpretare în lumea posibilă w_2, astfel ca formulele de sub operatorul cu asteriscul sus să-şi păstreze în w_2 valoarea pe care au avut-o în w_1, în timp ce valoarea formulei de sub operatorul cu asteriscul jos şi-a schimbat valoarea.

Exemple. a) Formula

$$f = \Box(p \vee \Diamond q) \to (\Box p \vee \Diamond q)$$

nu este validă.

Întocmim diagrama :

$$w_1 \quad \boxed{\begin{array}{l} \overset{*}{\Box}(p \vee \overset{*}{\Diamond} q) \to (\Box p \vee \overset{*}{\Diamond} q) \\ 1\ 1\ 1\ 0\ 0\ 0\ \ \underset{*}{0}\ \ 1\ 0\ 0\ 0 \end{array}}$$

$$\downarrow$$

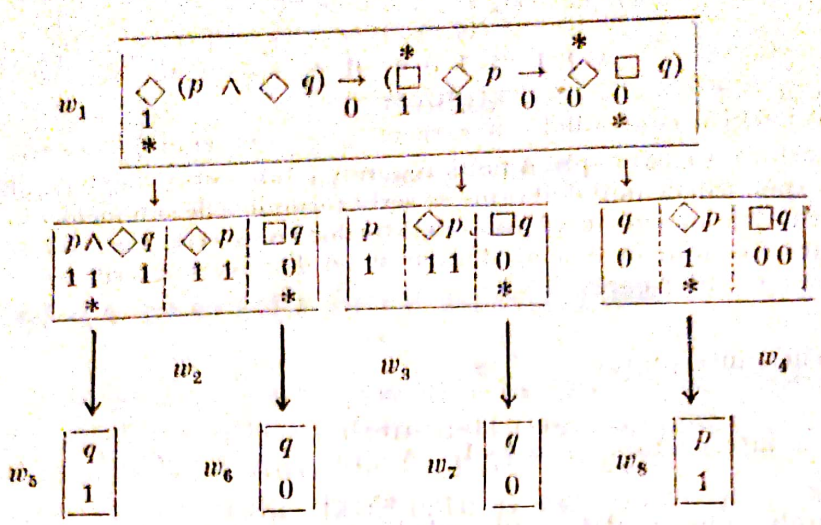
$$w_2 \quad \boxed{\begin{array}{c|c|c} p & p \vee \Diamond q & q \\ 0 & 0\ 1\ \underset{*}{1}\ 0 & 0 \end{array}}$$

$$\downarrow$$

$$w_3 \quad \boxed{\begin{array}{c} q \\ 1 \end{array}}$$

Formula a putut fi falsificată. Nu a apărut nici o contradicție.

b) Formula $f = \Diamond(p \wedge \Diamond q) \to (\Box \Diamond p \to \Diamond \Box q)$ nu este validă.



Nu există contradicție.

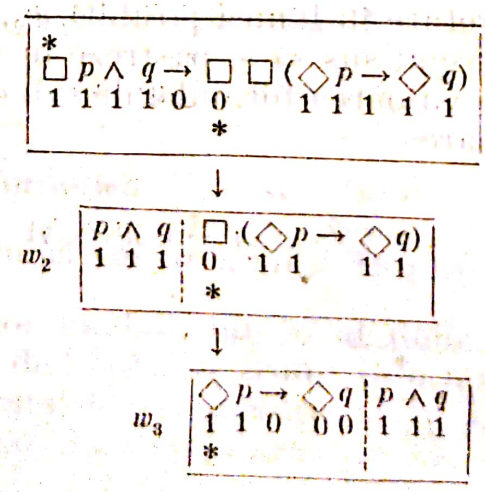
Formulele tratate pînă acum au aparținut sistemului **T**. În diagramele celorlalte sisteme se ține seama de regulile suplimentare:

S4: săgețile leagă un bloc de fiecare din blocurile succesive, numite *blocuri accesibile* (nu numai de următorul bloc ca în sistemul **T**), iar formulele sale de sub asteriscul de sus trebuie scrise și examinate în toate blocurile succesive.

S5: săgeata leagă fiecare bloc de fiecare bloc din diagramă. Cele două reguli reflectă cerința ca relația de accesibilitate între blocuri să fie tranzitivă în **S4** și, în plus, simetrică în **S5**.

Exemple. a) Să se examineze formula sistemului **S4**

$$\Box p \wedge q \to \Box\Box(\Diamond p \to \Diamond q)$$



Observăm că s-a obținut o contradicție în ultimul bloc ($q = 0$, $q = 1$) numai datorită faptului că am continuat să introducem $p \wedge q$ în al treilea bloc, deși formula $p \wedge q$ nu se găsește sub incidența unui operator modal în blocul 2. Prezența căsuței $p \wedge q$ în blocul 3 constituie diferența între procedeele celor două sisteme. Inconsistența apare în sistemul **S4**, dar nu și în sistemul **T**, deci formula este o teoremă **S4**, fără să fie teoremă în **T**.

b) Fie următoarea formulă validă a sistemului **S5**:

$$\Diamond\Diamond p \to \Box\Diamond p$$

Scanned with OKEN Scanner

Diagrama de decizie este reprezentată în fig. 8.2.

Inconsistența provine din legarea specifică în S5 a blocurilor, ceea ce face ca p să capete în w_4 valorile contradictorii 1, 0.

8.8. MODELELE LUI KRIPKE

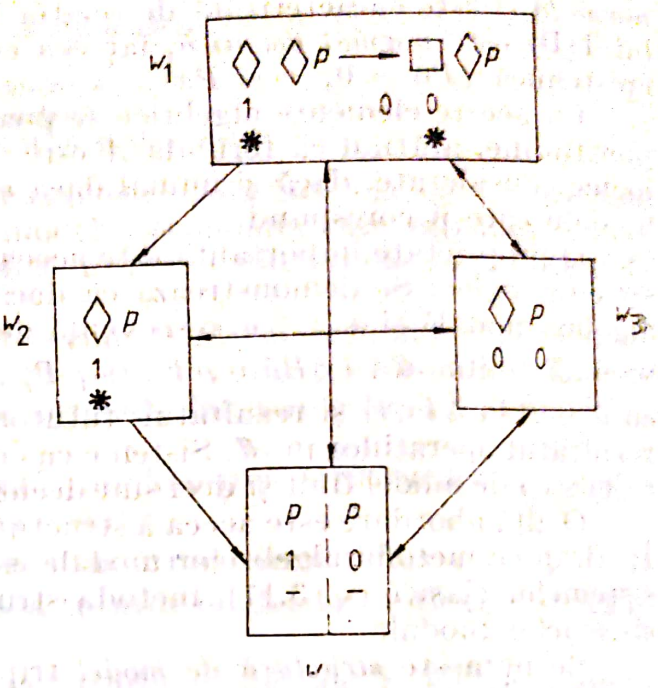


Fig. 8.2

Există posibilitatea asocierii unor algebre sistemelor logice modale, care au fost numite *algebre modale*. Spunem că $\mathcal{M} = \langle B, \cup, \cap, \sim, P\rangle$ este o algebră modală, dacă 1) B este o algebră booleană față de $\cup, \cap, \sim$ și dacă 2) pentru orice $x, y \in B$, $P(x \cup y) = Px \cup Py$, unde P este ca și $\sim$ un operator monadic. Se introduce prin definiție operația $Nx = \sim P \sim x$. În orice algebră modală :

1) pentru orice $x, y \in B$, $N(x \cap y) = Nx \cap Ny$;
2) pentru orice $x, y \in B$, dacă $x \leqslant y$, atunci $Px \leqslant Py$ și $Nx \leqslant Ny$.

Demonstrație. 1) $N(x \cap y) = \sim P \sim (x \cap y) = \sim P(\sim x \cup \sim y) =$
$= \sim(P \sim x \cup P \sim y) = \sim P \sim x \cap \sim P \sim y = Nx \cap Ny$.

2) $x \leqslant y$, $x \cup y = y$, $Px \cup Py = P(x \cup y) = Py$, $Px \leqslant Py$. La fel pentru $Nx \leqslant Ny$.

Se observă că P și N corespund operatorilor modali $\Diamond$ și $\Box$.

O algebră devine o *matrice* pentru o logică, cînd din mulțimea elementelor sale a fost desemnată o submulțime numită *mulțimea elementelor desemnate* (D). Unei algebre cu o mulțime de N elemente îi corespund 2^N matrice distincte. Interpretarea unei logici propoziționale într-o matrice constă în a face să corespundă fiecărei formule logice A cu n variabile o funcție f^A a matricei, construită din elementele ce corespund variabilelor propoziționale și operațiile algebrice ce corespund operațiilor logice. Formula A va fi satisfăcută de matrice dacă valoarea lui f^A pentru orice formație de n elemente se află în D. Altfel A este falsificată de D. Un sistem logic este satisfăcut de o matrice dacă toate teoremele sale sînt satisfăcute de matrice. O matrice se numește *matrice caracteristică a unui sistem logic* dacă și numai dacă formulele satisfăcute de matrice coincid cu teoremele sistemului.

O matrice $\langle B, D, \cup, \cap, \sim, P\rangle$ este *proprie* cînd $D \subset B$.

O matrice este *regulată* dacă : 1) este proprie, 2) D este un filtru al lui B ($x, y \in D$, $x \cap y \in D$; $x \in D$, $y \in B$, $x \cup y \in D$), 3) dacă $x \leftrightarrow y \in D$, atunci $x = y$, unde $x \leftrightarrow y = (x \rightarrow y) \cap (y \rightarrow x)$, iar $x \rightarrow y = \sim x \cup y$.

Dat fiind un sistem logic, se caută să se construiască matricea sa caracterisică. De exemplu, pentru sistemele C2, D2, E2, T(C), T(D), T

există matrice caracteristice regulate astfel că mulțimea elementelor desemnate conține un singur element ($d = 1$). Algebra modală care corespunde lui **D2** se numește *deontică* și este definită de relația $P1 = 1$. Algebra modală care corespunde lui **E2** se numește *epistemică* și este definită de relația $x \leqslant Px$. Algebra modală care corespunde lui **K** este numită *normală* și este caracterizată de relația $P0 = 0$. Algebra corespunzătoare lui **T(D)** este *normal deontică*, iar cea corespunzătoare lui **T** este normal epistemică ($P0 = 0$, $x \leqslant Px$).

Cu aceste elemente algebrice se poate demonstra o teoremă de completitudine, arătînd că formula A este o teoremă într-unul din sistemele logice considerate, dacă și numai dacă este satisfăcută de toate algebrele modale care îi corespund.

O proprietate importantă este aceea de *model finit* pe care o au unele sisteme logice. Se demonstrează că dacă $\mathcal{M} = \langle B, \cup, \cap, \sim, P\rangle$ este o algebră modală și $a_1 \ldots a_r$ o secvență de elemente din B, există o algebră modală finită $\dot{\mathcal{M}} = \langle \dot{B}, \cup, \cap, \sim, P\rangle$ cu cel mult $2^{2^{r+1}}$ elemente astfel ca $a_i \in \dot{B}$ $(1 \leqslant i \leqslant r)$ și rezultatul tuturor operațiilor în $\dot{\mathcal{M}}$ să coincidă cu rezultatul operațiilor în $\mathcal{M}$. Sisteme ca cele considerate în exemplu au proprietatea de model finit și deci sînt decidabile.

O altă abordare este aceea a structurilor de modele inițiată de Kripke. În timp ce metoda algebrelor modale este paralelă cu aceea a algebrelor sistemelor clasice (v. 3.11), metoda structurilor lui Kripke este specifică sistemelor modale.

Se numește *structură de model* tripletul $\langle W, R, V\rangle$, unde W este o mulțime nevidă, R o relație binară pe elementele acestei mulțimi și V o funcție de evaluare. Elementele mulțimii W se numesc *lumi posibile*. Evaluarea nu constă în atribuirea simplă unei valori unei variabile propoziționale, ca în calculul clasic, ci a unei valori într-o lume posibilă. V va fi o funcție de două variabile (variabila propozițională p și lumea posibilă $w \in W$). Folosirea diagramelor (v. 9.7) constituie o introducere intuitivă în această metodă a structurilor triplet.

Relația R se numește *relația de accesibilitate între lumi posibile*. Dacă wRw', unde $w, w' \in W$, spunem că w este în relație cu w' sau că w' este accesibilă din w.

Regulile care definesc pe V sînt următoarele:

1) Pentru orice formulă A și orice $w \in W$, $V(\sim A, w) = 1$ dacă $V(A, w) = 0$, altfel $V(\sim A, w) = 0$.

2) Pentru orice formule A, B și $w \in W$, $V(A \vee B, w) = 1$ dacă $V(A, w) = 1$ sau $V(B, w) = 1$; altfel $V(A \vee B, w) = 0$.

3) Pentru orice formulă A și $w \in W$, $V(\Box A, w) = 1$ dacă $V(A, w') = 1$ pentru orice $w' \in W$ astfel ca wRw'; altfel $V(\Box A, w) = 0$.

Regula 3) care face uz de relația de accesibilitate poate fi enunțată și pentru $\Diamond$:

3′) Pentru orice formulă A și $w \in W$, $V(\Diamond A, w) = 1$ dacă există cel puțin un w' astfel ca wRw' și $V(A, w') = 1$; altfel $V(\Diamond A, w) = 0$.

Validitatea unei formule A într-un model $\langle W, R, V\rangle$ este asigurată dacă și numai dacă $V(A, w) = 1$ pentru orice $w \in W$.

Vom schița succesiunea teoremelor care permit utilizarea metodei structurilor de model la studiul completitudinii sistemelor modale. Ne vom ocupa de sistemele care conțin pe **T** și care, ca și acest sistem, se numesc *normale* **(T, S4, B, S5)**.

Rezultatul cel mai important este că pentru studiul unor astfel de sisteme nu este nevoie să luăm în considerare toate structurile posibile, ci anumite clase de modele. Aceste clase sînt obținute prin impunerea unor anumite condiții relației de accesibilitate. Tabelul care urmează arată legătura între sistemele model normale și clasa de modele care le corespund.

Sistemul modal	Proprietatea lui R din modelul $\langle W, R, V\rangle$
T	R reflexivă
S4	R reflexivă și tranzitivă
B	R reflexivă și simetrică
S5	R reflexivă, tranzitivă și simetrică.

Analog cu matricea caracteristică care satisface exclusiv teoremele unui sistem modal, se introduce modelul economic care satisface teoremele sistemului, în timp ce falsifică într-una din lumile sale formulele care nu sînt teoreme.

Ca și în cazul clasic, faptul că teoremele sînt valide în model se demonstrează ușor, în timp ce reciproca, propunerile oricărei formule valide de a fi teoremă cere o demonstrație laborioasă, dar care este în linii mari cea folosită în cazul clasic (v. 2.7).

În final, teoremele care rezultă din folosirea structurilor de model Kripke pentru cele patru sisteme sînt:

1) **T** este complet față de clasa tuturor modelelor reflexive.

2) **S4** este complet față de clasa tuturor modelelor reflexive și tranzitive.

3) **B** este complet față de clasa tuturor modelelor reflexive și simetrice.

4) **S5** este complet față de clasa tuturor modelelor în care R este o relație de echivalență.

8.9. LOGICILE MODALE ȘI ALGEBRELE BOOLEENE TOPOLOGICE

Mersul asocierii unei algebre topologice unei logici modale este analog cu asocierea unei logici pseudobooleene unei logici intuiționiste (v. 6.2). Elemente de algebră topologică au fost introduse în 3.10. În 8.8 am considerat operația P drept fundamentală și N derivată. Este același lucru să procedăm invers.

Completăm alfabetul unei teorii de ordinul zero cu N numit *semnul necesității*. Dacă A este o formulă în limbajul acestei teorii, NA este de asemenea o formulă și se citește: „este necesar ca A". Prezența lui N face ca limbajul să se numească *modal*. Adoptăm și regula de inferență $\dfrac{A \to B}{NA \to NB}$. Formăm un sistem modal adăugînd la axiomele T1 — T12 (v. 3.11) și axiomele următoare:

M1 $NA \cap NB \to N(A \cap B)$ M3 $NA \to NNA$

M2 $NA \to A$ M4 $N(A \cup \sim A)$

O logică modală atribuie unui limbaj o operație de consecință modală, definită analog cu operația de consecință clasică, ținînd seama însă de noile axiome și noile reguli de inferență. Un calcul propozițional modal de ordinul zero va avea forma $\mathscr{P} = \{L, C_m\}$, iar o teorie modală de ordinul zero va avea forma $\mathscr{T} = \{L, C_m, \mathscr{A}\}$.

Toate sistemele definite în acest capitol sînt de această formă. Formulele din $C_m(0)$ sînt formule *modal derivabile*.

Algebra formulelor va fi $\langle F, \cup, \cap, \to, N\rangle$.

Relația $A \approx B$ care are loc numai dacă $A \to B$ și $B \to A$ sînt teoreme în T este o relație de congruență față de operatorii clasici. Ea este congruență și în raport cu N, deoarece dacă

$A \to B$ și $B \to A$ sînt teoreme, tot astfel sînt $NA \to NB$ și $NB \to NA$ conform regulii de inferență și deci $NA \approx NB$.

Algebra teoriei $\mathscr{T}$ este

$$\mathscr{A}(\mathscr{T}) = \langle F/\approx, \cup, \cap, \to, \sim, N \rangle$$

Rezultatul următor completează pe cele de la 7.11.

Algebra $\mathscr{A}(\mathscr{T})$ este o *algebră booleană topologică*. Pentru oricare formule A, B

$$|A| \cup |B| = |A \cup B|, \qquad |A| \cap |B| = |A \cap B|$$
$$|A| \to |B| = |A \to B|, \qquad \sim|A| = |\sim A| \text{ și } N|A| = |NA|$$

Pentru orice formulă A avem $|A| = 1$ dacă și numai dacă A este teoremă în $\mathscr{T}$.

Algebra booleană topologică $\mathscr{A}(\mathscr{T})$ este nedegenerată dacă și numai dacă $\mathscr{T}$ este consistentă.

Introducerea unei funcții de evaluare, definirea validității modale (tautologie propozițională modală) și a modelului adecvat (canonic) permit întreprinderea studiului consistenței și completitudinii sistemelor logice modale, folosind proprietățile algebrelor booleene topologice.

8.10. LOGICĂ MODALĂ ȘI LOGICĂ INTUIȚIONISTĂ

Între logica modală **S4** și logica intuiționistă Heyting a fost stabilit un raport exprimat în faptul că o formulă A este validă în sistemul Heyting dacă și numai dacă mulțimea formulelor de forma A este validă în **S4**. Trecerea de la un sistem la altul se face prin următoarea transcriere:

Logică intuiționistă	*Logică modală*
p	$\Box p$
$p \vee q$	$(p) \vee (q)$
$p \wedge q$	$(p) \wedge (q)$
$\sim p$	$\Box \sim p$
$p \to q$	$\Box(p \to q)$

Se observă că noțiunea intuiționistă „există construcție pentru p" devine în logica modală echivalentul necesarului. Implicația intuiționistă este implicația strictă. Elementul cheie al corespondenței este însă definiția negației ca imposibilitate: $\Box \sim$.

Gr. C. Moisil arată că dacă adăugăm unei logici pozitive (v. 6.4) axiomele imposibilității, notate de el η (v. 7.4) ca negație, obținem logica lui Heyting.

Aceste axiome sînt:

$$\vdash ((x \to y) \wedge (x \to \eta y)) \to \eta x$$
$$\vdash \eta x \to (x \to y).$$

Gr. C. Moisil a construit o logică modală generală bazată pe o logică pozitivă Hilbert-Bernays și pe un functor numit de el S, introdus printr-o axiomă și o regulă de inferență.

Notăm operatorul binar S cu $\diagdown$. El este introdus prin

M1 $\quad q \to (p \diagdown q) \vee p$ $\qquad$ M2 $\quad \dfrac{r \to p \vee q}{(r \diagdown q) \to p}$

Alegerea lui $\diagdown$ ca operator modal fundamental este justificată prin faptul că $p \diagdown q$ este dualul față de $\cup$, mai puțin utilizat al pseudocomple-

mentului relativ față de $\cap$: $p \to q$. Putem citi $p \setminus q$: „p fără q“ sau „p cînd non q“.

Din logica modală generală se deduc un mare număr de sisteme logice. În particular orice teză a logicii modale generale ce nu conține decît operatorii $\to$, $\vee$, $\wedge$ și $\sim$ (definită ca η) este o teză a logicii lui Heyting.

8.11. NOTĂ ISTORICĂ

Dintre toate logicile neclasice cea mai bogată istorie o are logica modală. Întrucît teoria silogismului cuprindea premise modale (este necesar ca p, este posibil ca p), modalitatea a fost intens studiată de Aristotel și a devenit apoi o temă recurentă în logica medievală. Propoziții care conțineau „necesarul“ au fost numite mai tîrziu *apodictice*, cele care conțineau „posibilul“ *problematice*, iar cele fără modalități *asertorice*. Definirea „necesarului“ și „posibilului” au ocupat o mare parte din preocupările școlii megarice și stoice.

Algoritmul scolastic modal reprezintă regulile conform cărora un enunț modal se poate transforma în mod corect într-altul. Cele patru modalități sînt : *possibile* μ, *contingens* γ, *impossibile* η, *necessarium* ν.

O propoziție modală este compusă din *dictum* p și *modus* (μ, γ, η sau ν). După cum este negată sau afirmată propoziția sau modul, există patru tipuri de enunțuri desemnate de literele E, I, A, U. Regula utilizării lor este rezumată de un vers : „*E* dictum negat, *I*que modum, nihil *A*, *U*totum“.

Să luăm exemplu $p \triangleq$ „Socrate fuge“. „Este posibil ca Socrate să nu fugă“ este un enunț de forma $\mu(\sim p)$, în care este negată propoziția și nu modalitatea, deci de tipul E. Cum traducem același enunț în celelalte modalități? Patru cuvinte ne dau dezlegarea : *PURPUREA*, *ILIACE*, *EDENTULI*, *AMABIMUS*. Cu alte cuvinte, dacă o propoziție este E în „possibile“, va fi E în „contingens“, U în „impossibile“ și I în „necessarium“ (*EdEntUlI*).

Iată tabloul care rezumă algoritmul :

	$\mu(p)$ POSSIBILE	$\gamma(p)$ CONTINGENS	$\eta(p)$ IMPOSSIBILE	$\nu(p)$ NECESSARIUM
PURPUREA	*U*	*U*	*E*	*A*
ILIACE	*I*	*I*	*A*	*E*
EDENTULI	*E*	*E*	*U*	*I*
AMABIMUS	*A*	*A*	*I*	*U*

Tabloul corespunde următoarelor formule în scrierea modernă (unde am utilizat bara superioară pentru negație) :

$$\begin{aligned}
\overline{\mu(\bar{p})} = \overline{\gamma(\bar{p})} &= \boxed{\overline{\gamma(p)}} = \eta(\bar{p}) = \nu(p) \\
\overline{\mu(p)} = \overline{\gamma(p)} &= \boxed{\overline{\gamma(\bar{p})}} = \eta(p) = \nu(\bar{p}) \\
\mu(\bar{p}) = \gamma(\bar{p}) &= \boxed{\gamma(p)} = \overline{\eta(\bar{p})} = \overline{\nu(p)} \\
\mu(p) = \gamma(p) &= \boxed{\gamma(\bar{p})} = \overline{\eta(p)} = \overline{\nu(\bar{p})}
\end{aligned}$$

Singura diferență față de logica modală de astăzi este identificarea la antici a contingentului cu posibilul. În coloana încadrată am pus versiunea modernă.

Vom ilustra cazul EDENTULI, urmărind linia a treia din ambele tablouri. Următoarele enunțuri sînt echivalente conform cu algoritmul scolastic :

Socrate non currere est possibile.
Socrate non currere est contingens.
Socrate non currere non est impossibile.
Socrate currere non est necesse.

În logica lui Teofil Coridaleu (1574—1646) după care se învăța în Academiile domnești din Țările Românești în secolele XVII și XVIII, există algoritmul modal de mai sus, cu termeni mnemotehnici greceşti.

După ce în secolul XVIII, calculul propozițiilor a fost axiomatizat, la începutul secolului nostru au început încercările de a-l extinde în direcția înglobării „necesarului" și „posibilului". Primele sugestii apar la H. MacColl (1906), dar prind corp o dată cu lucrarea lui C. I. Lewis „Survey of Symbolic Logic" din 1918. Lewis ajunge la logica modală din revolta sa împotriva implicației materiale, care stătea la baza calculului clasic aprofundat în „Principia Mathematica". Implicația strictă pentru care el se pronunță este $p \prec q \equiv \Box(p \to q)$. Cele cinci sisteme ale lui Lewis și elevilor săi, **S1 — S5,** elaborate pe parcursul anilor, constituie o construcție remarcabilă : sistemul **S1** este caracterizat de axioma $p \to \Diamond p$ ($\Diamond$ este operatorul primar la Lewis) ; prin adăugarea axiomei $\Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$ se obține **S2;** prin adăugarea la acesta a axiomei $p \to q \to (\Diamond p \to \Diamond q)$ rezultă **S3;** dacă se adaugă $\Box p \to \Box\Box p$, avem **S4;** în sfîrșit, prin suplimentarea cu $\Diamond p \to \Box\Diamond p$ se obține **S5.** Următoarea cronologie sumară privind sisteme citate în acest capitol este sugestivă : sistemele **S4** și **S5** (Lewis 1932), sistemul **T** (Feys și Von Wright 1950, 1951), sistemul **B** (Kripke 1954), sistemul **S4.3** (Dummett și Lemmon 1959), sistemul **D** (Kripke 1965).

Una din direcțiile de dezvoltare cele mai fertile ale logicii modale o constituie *logica deontică* și diferitele variante ale *logicii acțiunii*. Inițiată de von Wright, logica deontică sau *logica normelor*, aprofundată în numeroase studii în țara noastră, și-a găsit aplicații, începînd cu domeniul juridic.

A doua direcție o reprezintă logica temporală și diferitele variante ale logicii programelor, de mare actualitate în aplicațiile în informatică și inteligență artificială.

Ceea ce caracterizează cercetările prezente este studiul și aprofundarea rezultatelor logicii modale, dobîndite anterior în perspectiva teoretică și filozofică, în vederea unor aplicații de o capitală importanță.

8.12. EXERCIȚII ȘI PROBLEME

1. Să se demonstreze în sistemul T :

a) $\vdash \Diamond(p \to q) \equiv \Box p \to \Diamond q$,

b) $\vdash \Diamond \sim p \vee \Diamond \sim q \vee \Diamond(p \vee q)$,

c) $\vdash \Diamond(p \to (q \wedge r)) \to ((\Box p \to \Diamond q) \wedge (\Box p \to \Diamond r))$.

Scanned with OKEN Scanner

2. Să se demonstreze în **T**:

$$\vdash(A \to B) \to \vdash(XA \to XB),$$

unde X este o modalitate afirmativă (o secvență de $\Box$, $\Diamond$ și un număr par de $\sim$).

3. Dacă X este o modalitate afirmativă, să se demonstreze în **S4** că următoarea formulă este o teoremă:

$$p \prec q \to (Xp \prec Xq).$$

4. Să se stabilească în sistemul **B** următoarea regulă derivată:

$$\frac{\Diamond A \to B}{A \to \Box B}$$

5. Să se reducă la forma FNCM următoarele formule:

a) $\Box(\sim p \wedge \sim q) \to [\Box(p \vee q) \prec r) \wedge (r \to (p \prec p))]$,

b) $(p \prec (q \prec (p \prec q))) \to (\sim(p \prec q) \to (p \prec \sim q))$.

6. Să se găsească celelalte șapte modalități distincte ale lui **S4** și să se arate grafic relația dintre ele (v. 8.4).

7. Regula $\dfrac{A \to \Box(B \to C)}{B \to \Box(A \to C)}$ nu conservă adevărul în **T**. Dați un exemplu.

8. Să se aplice metoda diagramelor semantice pentru studiul validității formulelor din **T**:

a) $\Diamond(p \wedge q) \vee \Diamond(p \wedge r) \to \Box p$,

b) $\Box(p \equiv q) \to \Box(\Box p \equiv \Box q)$.

9. Cercetați prin diagrame validitatea formulelor din **S4**

a) $\Box(\Box p \to q) \vee \Box(\Box q \to p)$,

b) $\Box(\Box p \to q) \to \Box(\Box p \to \Box q)$.

10. Cercetați prin diagrame validitatea formulelor din **S5**

a) $\Box(p \vee (q \vee (r \vee \Box s)))$,

b) $\Diamond(p \wedge q) \to ((\Box p \prec \Box q) \prec \Diamond q))$.

11. Să se arate că adăugînd la **T** axioma

$$p \prec \Box\Box\Diamond p$$

se obține **S5**.

12. Să se înlocuiască axioma A0 în sistemul **T** cu axioma mai puternică

$$\Box(p \to q) \to \Box(\Box p \to \Box q).$$

Să se arate că sistemul astfel obținut este echivalent cu **S4**.

13. Sistemul obținut din **S4** prin adăugarea axiomei

$$\Diamond\Box p \to \Box\Diamond p$$

se numește **S4.2**. Să se arate că: a) **S4.2** este conținut în **S5**; b) **S4.3** conține **S4.2**; c) **S4.2** are zece modalități neechivalente, din care cele cinci pozitive sînt reprezentabile în tabloul

$$\begin{array}{ccccc} \Diamond\Box p & \longrightarrow & & & \Box\Diamond p \\ & \nearrow & & \searrow & \\ \Box p & \longrightarrow & p & \longrightarrow & \Diamond p \end{array}$$

14. Să se arate că adăugînd la **S4** axioma

$$\Box\Diamond p \to \Diamond\Box p$$

se obține un sistem care nu este cuprins în **S5**.

15. Să se considere cele șase axiome ale sistemului **S1** :

A1 $(p \cdot q) \prec (q \cdot p)$
A2 $(p \cdot q) \prec p$
A3 $p \prec (p \cdot p)$
A4 $((p_1 \cdot q) \cdot r) \prec (p \cdot (q \cdot r))$
A5 $(p \prec q) \cdot (q \prec r) \prec (p \prec r)$
A6 $p \cdot (p \prec q) \prec q$

cu regulile de substituție, MP și adjuncție $\dfrac{\vdash A,\ \vdash B}{\vdash A \cdot B}$. Să se arate că adăugînd axioma $p \prec \Box \Diamond p$ (axioma lui Brouwer strictă) la **S1**, se obține **B**. Se poate obține **S5** prin adăugarea la S1 a axiomei $p \prec \Box\Box \Diamond p$?

16. Să se arate că toate algebrele epistemice sînt deontice (v.8.8).

BIBLOIGRAFIE

1. Chellas, B. F. *Modal Logic : An Introduction.* Cambridge University Press, 1980.
2. Corydalée, T. *Introduction à la logique* (tradus și prezentat de C. Noica). București, Editura Academiei R.S.R., 1970.
3. Feys, R. *Modal Logics.* Nauwelaerto, Louvain, 1965.
4. Grigolia, R. S. *Svobodnie S4.3 alghebri s konecinim cislom obrazuiuşcih. Issledovania po neklassiceskim loghikam i formalnim sistemam.* Moskova, Nauka, 1983.
5. Hughes, G. E., Cresswell, M. J. *An Introduction to Modal Logic.* London, Methuen, 1972.
6. Hughes, G. E., Cresswell, M. J. *A Companion to Modal Logic.* London, Methuen, 1984.
7. Ivlev, M. V. *Soderjatelnaia semantika modalnoi loghiki.* Izd. Moskovskogo Universiteta, 1985.
8. Lemmon, E. J. *Algebraic Semantics for Modal Logic.* I—II. In : Journal of Symbolic Logic, 31, nr. 1, March, 1966, nr. 2, June, 1966.
9. Moisil, Gr. C. *Logica modală. Încercări vechi și noi de logică clasică.* București, Editura Științifică, 1965.
10. Popa, C. *Teoria acțiunii și logica formală.* București, Editura Științifică și Enciclopedică, 1984.
11. Popa, C. (coord.) *Logica acțiunii. Studii.* București, Editura Științifică și Enciclopedică, 1983.
12. Rața, M. F. *O funkționalnoi polnote v modalnoi loghike S5. Issledovnaia po neklassiceskim loghikan i formalnim sistemam.* Moskova, Nauka, 1983.
13. Schütte, K. *Vollständige Systeme modaler und intuitionistischer Logik.* Berlin, Springer Verlag, 1968.
14. Segerberg, K. *An Essay in Classical Modal Logic* (3 vol.). Uppsala, Filosofizica Studier.
15. Von Wright, G. H. *Normă și acțiune.* București, Editura Științifică și Enciclopedică, 1982.
16. Zeman, J. J. *Modal Logic : The Lewis Systems.* Oxford, Clarendon Press, 1973.

Scanned with OKEN Scanner

9.
INTRODUCERE ÎN LOGICA TEMPORALĂ

9.1. O LOGICĂ TEMPORALĂ CU ○

Logica temporală este o logică modală urmărind să exprime procesele ce se desfășoară în timp. La fiecare interpretare se ține seama de starea în care se află procesul, dintr-un șir infinit de stări :

$$s_0, s_1, s_2, \ldots$$

Ca și în cazul logicilor polivalente și modale ne vom ocupa numai de calculul propozițiilor într-o asemenea logică. Operatorii modali sînt următorii :

$\Box p$ totdeauna p

$\Diamond p$ cîndva p

care joacă rolul necesarului și posibilului și noii operatori pe care îi numim

$\bigcirc p$ p în următorul (moment)

$p \Delta q$ p apoi q

$\Box$, $\Diamond$, și $\bigcirc$ sînt operatori unari, iar Δ este operator binar.

Vom da următoarea interpretare intuitivă expresiilor cu operatori :

$\Diamond \Box p$: cîndva p va deveni permanent adevărat ;

$p \to \Diamond q$: dacă p este adevărat acum, q va fi adevărat cîndva (în viitor) ;

$\Diamond(p \wedge \bigcirc \sim p)$: într-un moment viitor (cîndva), p va fi adevărat și în următorul moment va fi fals ;

$\Box \Diamond p$: orice moment (totdeauna) este urmat de perspectiva ca p să fie adevărat ;

$p \Delta q$: după ce p este adevărat (un anumit timp), q devine adevărat (p, apoi q) ;

$\Box p \vee (p \Delta q)$: sau p este totdeauna adevărat, sau este adevărat pînă la apariția lui q (cînd q devine adevărat) ;

$\Diamond p \to (\sim q \Delta p)$: dacă p se întîmplă (devine adevărat) cîndva, atunci aceasta se produce o dată cu apariția lui q (încetarea lui $\sim q$).

Vom păstra sensul și utilizarea celorlalte simboluri ale CP ca $\models$ validitate (logică), $\vdash$ teoremicitate etc.

Scanned with OKEN Scanner

Axiomele care definesc sistemul temporal propoziţional sînt :

A1 $\sim \Diamond p \equiv \Box \sim p$
A2 $\Box(p \to q) \to (\Box p \to \Box q)$
A3 $\Box p \to p$
A4 $\bigcirc \sim p \equiv \sim \bigcirc p$
A5 $\bigcirc(p \to q) \to (\bigcirc p \to \bigcirc q)$
A6 $\Box p \to \bigcirc p$
A7 $\Box p \to \bigcirc \Box p$
A8 $\Box(p \to \bigcirc p) \to (p \to \Box p)$
A9 $p \,\Delta\, q \equiv (q \vee (p \wedge \bigcirc(p \,\Delta\, q)))$
A10 $p \,\Delta\, q \to \Diamond q$.

Regulile de transformare sînt :

R1 Dacă p este tautologie, atunci este teoremă $\vdash p$.

R2 Modus ponens. Dacă $\vdash p \to q$ şi $\vdash p$, atunci $\vdash q$.

R3 Dacă $\vdash p$, atunci $\vdash \Box p$ (regula inserţiei).

Se observă că nu se utilizează regula substituţiei, întrucît p, q, r, ... sînt considerate scheme (de formule).

Vom prezenta teoreme (notate T) şi noi reguli derivate (notate R), care vor facilita demonstraţiile. Ca şi pînă acum, vom demonstra o parte din teoreme cu titlu ilustrativ.

R4 Dacă $\vdash (p_1 \wedge p_2 \wedge \ldots \wedge p_n) \to q$, $\vdash p_1$, $\vdash p_2$, ..., $\vdash p_n$, atunci $\vdash q$.

Demonstraţie. a) $\vdash (p_1 \wedge p_2 \wedge \ldots \wedge p_n) \to (p_1 \to (p_2 \to (\ldots \ldots p_n \to q))) \ldots)$ CP.

b) Se aplică MP de $n + 1$ ori.

R5 Dacă $\vdash p$, atunci $\vdash \bigcirc p$.

Demonstraţie. a) $\vdash p$,
b) $\vdash \Box p$ prin R3
c) $\vdash \bigcirc p$ prin A6, MP

T1 $\vdash p \to \Diamond p$

Demonstraţie. a) $\vdash \Box \sim p \to \sim p$ A3
b) $\vdash p \to \sim \Box \sim p$ CP
c) $\vdash p \to \Diamond p$ A1

R6 Dacă $\vdash u$, atunci $\vdash \Diamond u$.

T2 $\vdash \bigcirc p \to \Diamond p$

R7 1. Dacă $\vdash p \to q$, atunci $\vdash \Box p \to \Box q$.
2. Dacă $p \equiv q$, atunci $\vdash \Box p \equiv \Box q$.

Demonstraţie. a) $\vdash p \to q$
b) $\vdash \Box(p \to q)$ R3
c) $\vdash \Box(p \to q) \to (\Box p \to \Box q)$ A2
d) $\vdash \Box p \to \Box q$ MP

R8 1. Dacă $\vdash p \to q$, atunci $\vdash \Diamond p \to \Diamond q$.
2. Dacă $p \equiv q$, atunci $\vdash \Diamond p \equiv \Diamond q$.

Scanned with OKEN Scanner

R9 1. Dacă $\vdash p \to q$, atunci $\vdash \bigcirc p \to \bigcirc q$.
2. Dacă $\vdash p \equiv q$, atunci $\vdash \bigcirc p \equiv \bigcirc q$.

R10 Dacă $\vdash p \to \bigcirc p$, atunci $\vdash p \to \Box p$.

Demonstrație. a) $\vdash p \to \bigcirc p$
b) $\vdash \Box(p \to \bigcirc p)$ R3
c) $\vdash \Box(p \to \bigcirc p) \to (p \to \Box p)$ A8
d) $\vdash p \to \Box p$ MP

R11 Dacă $\vdash p \to (q \wedge \bigcirc p)$, atunci $\vdash p \to \Box q$

Demonstrație. a) $\vdash p \to (q \wedge \bigcirc p)$
b) $\vdash p \to \bigcirc p$ R4
c) $\vdash p \to \Box p$ R10
d) $\vdash p \to q$ R4
e) $\vdash \Box p \to \Box q$ R7
f) $\vdash p \to \Box q$ MP

T3 $\vdash \Box p \equiv \Box\Box p$
T4 $\vdash \Diamond p \equiv \Diamond\Diamond p$
T5 $\vdash \Diamond \sim p \equiv \sim \Box p$
T6 $\vdash \Box(p \to q) \to (\Diamond p \to \Diamond q)$
T7 $\vdash \Box(p \wedge q) \equiv \Box p \wedge \Box q$
T8 $\vdash \Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$
T9 $\vdash \Box p \vee \Box q \to \Box(p \vee q)$
T10 $\vdash \Diamond(p \wedge q) \to \Diamond p \wedge \Diamond q$
T11 $\Box p \wedge \Diamond q \to \Diamond(p \wedge q)$
T12 $\bigcirc(p \wedge q) \equiv \bigcirc p \wedge \bigcirc q$

Demonstrație. a) $p \to (q \to (p \wedge q))$ CP
b) $\bigcirc p \to \bigcirc(q \to (p \wedge q))$ R9
c) $\bigcirc(q \to (p \wedge q)) \to (\bigcirc q \to \bigcirc(p \wedge q))$ A5
d) $\bigcirc p \wedge (\bigcirc q \to \bigcirc(p \wedge q))$ b), c)
e) $\bigcirc p \wedge \bigcirc q \to \bigcirc(p \wedge q)$ R4
f) $p \wedge q \to p$ CP
g) $\bigcirc(p \wedge q) \to \bigcirc p$ R9
h) $p \wedge q \to q$ CP
i) $\bigcirc(p \wedge q) \to \bigcirc q$ R9
j) $\bigcirc(p \wedge q) \to \bigcirc p \wedge \bigcirc q$ g), i)
k) $\bigcirc(p \wedge q) \equiv \bigcirc p \wedge \bigcirc q$ e), j)

T13 $\bigcirc(p \vee q) \equiv \bigcirc p \vee \bigcirc q$
T14 $\bigcirc(p \to q) \equiv \bigcirc p \to \bigcirc q$
T15 $\bigcirc(p \equiv q) \equiv (\bigcirc p \equiv \bigcirc q)$
T16 $\bigcirc\Box p \equiv \Box \bigcirc p$
T17 $\bigcirc\Diamond p \equiv \Diamond \bigcirc p$
T18 $\Box\Diamond\Box p \equiv \Diamond \Box p$

Scanned with OKEN Scanner

Demonstrație. a) $\Box\Diamond\Box p \to \Diamond\Box p$ A3
b) $\Box p \to \bigcirc\Box p$ A7
c) $\Diamond\Box p \to \Diamond\bigcirc\Box p$ R8
d) $\Diamond\Box p \to \bigcirc\Diamond\Box p$ T17
e) $\Diamond\Box p \to \bigcirc\Diamond\Box p$ c), d)
f) $\Diamond\Box p \to \Box\Diamond\Box p$ R10
g) $\Box\Diamond\Box p \equiv \Diamond\Box p$ a), f)

T19 $\Diamond\Box\Diamond p \equiv \Box\Diamond p$

T20 $\Box p \equiv (p \wedge \bigcirc\Box p)$

Demonstrație. a) $\Box p \to p$ A3
b) $\Box p \to \bigcirc\Box p$ A7
c) $\Box p \to (p \wedge \bigcirc\Box p)$ a), b)
d) $\bigcirc\Box p \to \bigcirc(p \wedge \bigcirc\Box p)$ R9
e) $p \wedge \bigcirc\Box p \to \bigcirc(p \wedge \bigcirc\Box p)$ R4
f) $p \wedge \bigcirc\Box p \to \Box(p \wedge \bigcirc\Box p)$ R10
g) $\Box(p \wedge \bigcirc\Box p) \to \Box p$ R7
h) $p \wedge \bigcirc\Box p \to \Box p$ f), g)
i) $\Box p \equiv p \wedge \bigcirc\Box p$ c), h)

T21 $\Diamond p \equiv p \vee \bigcirc\Diamond p$

T22 $(p \wedge \Diamond \sim p) \to \Diamond(p \wedge \bigcirc \sim p)$

R12 1. Dacă $\vdash p \to q$, $\vdash q \to \Box r$, $\vdash r \to s$, atunci $\vdash p \to \Box s$.
2. Dacă $\vdash p \to q$, $\vdash q \to \Diamond r$, $\vdash r \to s$, atunci $\vdash p \to \Diamond s$.
3. Dacă $\vdash p \to q$, $\vdash q \to \bigcirc r$, $\vdash r \to s$, atunci $\vdash p \to \bigcirc s$.

R13 1. Dacă $\vdash p \to \Box q$, $\vdash q \to \Box r$, atunci $\vdash p \to \Box r$.
2. Dacă $\vdash p \to \Diamond q$, $\vdash q \to \Diamond r$, atunci $\vdash p \to \Diamond r$.

Demonstrație. a) $p \to \Box q$
b) $q \to \Box r$
c) $\Box q \to \Box\Box r$ R7 și b)
d) $\Box q \to \Box r$ T3
e) $p \to \Box r$ R3

R14 Dacă $\vdash r \to \Diamond q$ și $\vdash r \to (q \vee (p \wedge \bigcirc r))$, atunci
$\vdash r \to (p \,\Delta\, q)$

Demonstrație. a) $\vdash r \to \Diamond q$
b) $\vdash r \to (q \vee (p \wedge \bigcirc r))$
c) $(q \vee (p \wedge \bigcirc(p \Delta q))) \to (p \Delta q)$ A9
d) $\sim(p \Delta q) \to (\sim q \wedge (\sim p \vee \bigcirc \sim(p \,\Delta\, q)))$ c), A4
e) $(r \wedge \sim(p \,\Delta\, q)) \to (\sim q \wedge \bigcirc r \wedge \bigcirc \sim(p \,\Delta\, q))$ b), d)
f) $(r \wedge \sim(p \,\Delta\, q)) \to (\sim q \wedge \bigcirc(r \wedge \sim(p \,\Delta\, q)))$ T12
g) $(r \wedge \sim(p \Delta\, q)) \to \Box \sim q$ R11
h) $r \wedge (\sim p \,\Delta\, q) \to \sim\Box \sim q$ a), T5
i) $w \to (p \Delta q)$ g), h)

Scanned with OKEN Scanner

R15 Dacă $\vdash (q \vee (p \wedge \bigcirc r)) \to r$, atunci $\vdash (p \,\Delta\, q) \to r$.

R16 1. Dacă $\vdash p \to q$, și $\vdash r \to s$, atunci $\vdash p \,\Delta\, q \to r \,\Delta s$.

2. Dacă $\vdash p \equiv q$ și $\vdash r \equiv s$, atunci $\vdash p \,\Delta\, q \equiv r \,\Delta\, s$.

T23 $\vdash (\sim p) \Delta p \equiv \Diamond p$

Demonstrație.

a) $(\sim p) \,\Delta\, p \to \Diamond p$ — A10

b) $\Diamond p \to (p \vee \bigcirc \Diamond p)$ — T21

c) $\Diamond p \to (p \vee (\sim p \wedge \bigcirc \Diamond p)$ — b)

d) $\Diamond p \to \Diamond p$

e) $\Diamond p \to (\sim p) \Delta p$ — c), d) și R14

f) $(\sim p) \,\Delta\, p \equiv \Diamond p$ — a), e)

T24 $\vdash (\Box p \wedge \Diamond q) \to (p \,\Delta\, q)$

Demonstrație.

a) $(\Box p \wedge \Diamond q) \to \Diamond q$

b) $(\Box p \wedge \Diamond q) \to (p \wedge \bigcirc \Box p) \wedge (q \vee \bigcirc \Diamond q)$ — T20, T2

c) $(\Box p \wedge \Diamond q) \to (q \vee (p \wedge \bigcirc \Box p \wedge \bigcirc \Diamond q))$ — b)

d) $(\Box p \wedge \Diamond q) \to (\sim q \vee (p \wedge \bigcirc (\Box p \wedge \Diamond q)))$ — T12

e) $(\Box p \wedge \Diamond q) \to p \,\Delta\, q$ — a), d), R14

T25 $\vdash (p \,\Delta\, q) \,\Delta\, q \equiv p \,\Delta\, q$

T26 $\vdash p \,\Delta\, q \equiv p \,\Delta\, (p \,\Delta\, q)$

R17 1. Dacă $\vdash v$, atunci $\vdash u \,\Delta\, v$.

2. Dacă $\vdash u$ și $\vdash \Diamond v$, atunci $\vdash u \,\Delta\, v$.

Demonstrație.

a) $\vdash u$

b) $\vdash \Diamond v$

c) $\vdash \Box u$ — a), R3

d) $\vdash (\Box u \wedge \Diamond v) \to u \,\Delta\, v$ — T24

e) $\vdash u \,\Delta\, v$ — b), c), d)

R18 Dacă $\vdash p \to s \,\Delta q$, $\vdash q \to s \,\Delta\, r$, atunci $\vdash p \to s \,\Delta\, r$.

Demonstrație.

a) $p \to s \,\Delta\, q$

b) $q \to s \,\Delta\, r$

c) $s \,\Delta\, q \to s \,\Delta (s \,\Delta\, r)$ — R16

d) $p \to s \,\Delta\, (s \,\Delta\, r)$ — a, c)

e) $p \to s \,\Delta\, r$

T27 $(\Box p \wedge q \,\Delta\, r) \to (p \wedge q) \,\Delta\, (p \wedge r)$

Demonstrație.

a) $q \,\Delta\, r \to \Diamond r$ — A10

b) $\Box p \wedge q \,\Delta\, r \to \Box p \wedge \Diamond r$ — a)

c) $\Box p \wedge q \,\Delta\, r \to \Diamond (p \wedge r)$ — T11

d) $q \,\Delta\, r \to (r \vee (q \wedge \bigcirc (q \,\Delta\, r)))$ — A9

e) $\Box p \wedge q \,\Delta\, r \to (\Box p \wedge r) \vee (\Box p \wedge q \wedge \bigcirc (q \,\Delta\, r)))$

f) $\Box p \wedge r \to p \wedge r$ — A3

g) $(\Box p \wedge q \wedge \bigcirc (q \,\Delta\, r)) \to (p \wedge q \wedge \bigcirc \Box p \wedge \bigcirc (q \Delta r))$ — T20

h) $\Box p \wedge q \wedge \bigcirc (q \,\Delta\, r) \to ((p \wedge q) \wedge \bigcirc (\Box p \wedge (q \,\Delta\, r)))$ — T12

Scanned with OKEN Scanner

i) $\Box p \wedge (q \,\Delta\, r) \to (p \wedge r) \vee ((p \wedge q) \wedge \bigcirc(\Box p \wedge (q \,\Delta\, r)))$ e), f), h)

j) $(\Box p \wedge q \,\Delta\, r) \to (p \wedge q) \,\Delta\, (p \wedge r)$ c), i), R14

T28 $\vdash (\bigcirc p) \,\Delta\, (\bigcirc q) \equiv \bigcirc(p \,\Delta\, q)$

T29 $\vdash (p \wedge q) \,\Delta\, r \equiv (p \,\Delta\, r \wedge q \,\Delta\, r)$

Demonstrație. a) $p \wedge q \to p$

b) $(p \wedge q) \,\Delta\, r \to p \,\Delta\, r$ R16

c) $(p \wedge q) \,\Delta\, r \to q \,\Delta\, r$ R16

d) $(p \wedge q) \,\Delta\, r \to (p \,\Delta\, r \wedge q \,\Delta\, r)$ b, c)

e) $p \,\Delta\, r \to \Diamond r$ A10

f) $p \,\Delta\, r \wedge q \,\Delta\, r \to \Diamond r$ e)

g) $p \,\Delta\, r \to r \vee (p \wedge \bigcirc(p \,\Delta\, r))$ A9

h) $q \,\Delta\, r \to r \vee (q \wedge \bigcirc(q \,\Delta\, r))$ A9

i) $p \,\Delta\, r \wedge q \,\Delta\, r \to r \vee ((p \wedge q) \wedge \bigcirc(p \,\Delta\, r \wedge q \,\Delta\, r)$ g), h), T12

j) $p \,\Delta\, r \wedge q \,\Delta\, r \to (p \wedge q) \,\Delta\, r$ f), i), R14

k) $(p \wedge q) \,\Delta\, r \equiv (p \,\Delta\, r) \wedge (q \,\Delta\, r)$ d), j)

T30 $\vdash p \,\Delta\, (q \vee r) \equiv (p \,\Delta\, q) \vee (p \,\Delta\, r)$

T31 $\vdash \Diamond p \vee \Diamond q \to ((\sim p) \,\Delta\, q \vee (\sim q) \,\Delta\, r)$

Demonstrație. a) $\Diamond p \vee \Diamond q \to \Diamond(p \vee q)$ T8

b) $\Diamond(p \vee q) \to (\sim(p \vee q)) \,\Delta\, (p \vee q)$ T23

c) $\Diamond(p \vee q) \to (\sim p \wedge \sim q) \,\Delta\, (p \vee q)$ R16

d) $\Diamond(p \vee q) \to (\sim p \wedge \sim q) \,\Delta\, p \vee (\sim p \wedge \sim q) \,\Delta\, q)$ T30

e) $(\sim p \wedge \sim q) \,\Delta\, p \to (\sim q) \,\Delta\, p$ R16

f) $(\sim p \wedge \sim q) \,\Delta\, q \to (\sim p) \,\Delta\, q$ R16

g) $\Diamond(p \vee q) \to ((\sim p) \,\Delta\, q \vee (\sim q) \,\Delta\, p)$ d), e), f)

h) $\Diamond p \vee \Diamond q \to ((\sim p) \,\Delta\, q \vee (\sim q) \,\Delta\, p)$ a), g)

T32 $\vdash p \,\Delta\, (q \wedge r) \to p \,\Delta\, q \wedge p \,\Delta\, r$

Demonstrație. a) $q \wedge r \to q$

b) $p \,\Delta\, (q \wedge r) \to p \,\Delta\, q$ R16

c) $p \,\Delta\, (q \wedge r) \to p \,\Delta\, r$ R16

d) $p \,\Delta\, (q \wedge r) \to p \,\Delta\, q \wedge p \,\Delta\, r$ b), c)

T33 $\vdash (p \,\Delta\, r \vee q \,\Delta\, r) \to (p \vee q) \,\Delta\, r$

T34 $(p \to q) \,\Delta\, r \to (p \,\Delta\, r \to q \,\Delta\, r)$

T35 $p \,\Delta\, q \wedge (\sim q) \,\Delta\, r \to p \,\Delta\, r$

Demonstrație. a) $(\sim q) \,\Delta\, r \to \Diamond r$ A10

b) $p \,\Delta\, q \wedge (\sim q) \,\Delta\, r \to \Diamond r$ a)

c) $p \,\Delta\, q \to (q \vee (p \wedge \bigcirc (p \,\Delta\, q)))$ A9

d) $(\sim q) \,\Delta\, r \to (r \vee (\sim q \wedge \bigcirc((\sim q) \,\Delta\, r)))$ A9

e) $p \,\Delta\, q \wedge (\sim q) \,\Delta\, r \to (r \vee (p \wedge \sim q \wedge \bigcirc(p \,\Delta\, q) \wedge \wedge \bigcirc((\sim q) \,\Delta\, r)))$ c), d)

Scanned with OKEN Scanner

f) $p \Delta q \wedge (\sim q) \Delta r \to (r \vee (p \wedge \bigcirc(p \Delta q \wedge (\sim q) \Delta r)))$ T12

g) $(p \Delta q \wedge (\sim q) \Delta r) \to p \Delta r$

T36 $p \Delta (q \wedge r) \to (p \Delta q) \Delta r$

T37 $(p \Delta q) \Delta r \to (p \vee q) \Delta r$

T38 $p \Delta (q \Delta r) \to (p \vee q) \Delta r$

Comentariu. Axiomele A1—A3 asigură un sistem modal de tipul **T**, prima fiind definiția operatorului $\Diamond$. Putem să le citim în felul următor: p este întotdeauna fals dacă și numai dacă nu va fi cîndva adevărat (A1); dacă p implică q întotdeauna, atunci p este permanent adevărat, q este întotdeauna adevărat (A2); dacă p este întotdeauna adevărat, este adevărat și în prezent. Axioma A4 arată că operatorul $\bigcirc$ este autodual, iar A5 este analogul lui A2 pentru operatorul $\bigcirc$. Axioma A6 spune că „întotdeauna" implică „următorul" (moment). Axioma A7 poate fi citită astfel: dacă p este adevărată întotdeauna, este adevărată în toate momentele care urmează următorului.

Avem și afirmația că o proprietate ce se păstrează întotdeauna de la un moment la următorul se păstrează permanent (A8). Axioma A9 definește operatorul binar Δ astfel: p apoi q înseamnă că în prezent sau (deja) q sau p, următorul fiind p apoi q. Iar A10 spune că p apoi q implică faptul că q se va întîmpla cîndva.

Regulile 3, 5, 6 permit inserarea operatorilor modali $\Box$, $\Diamond$, $\bigcirc$ în fața oricărei teoreme $\vdash p$. Regulile 7, 8, 9, 16 arată că toți operatorii $\Box$, $\Diamond$, $\bigcirc$, Δ sînt monotoni, în sensul că orice expresie aflată sub semnul lor poate fi înlocuită printr-alta mai slabă.

Nu există opratori iterați $\Box\Box \ldots$, $\Diamond\Diamond \ldots$ la fel ca în sistemul **S4** (T3 și T4). Demonstrația folosește proprietățile operatorului $\bigcirc$. Este evident că acesta nu se absoarbe: vom avea $\bigcirc$, $\bigcirc\bigcirc$, $\ldots$, $\bigcirc^k$ operatori distincți (următorul următorului nu este egal cu următorul). Dar $\bigcirc$ are proprietatea de a comuta cu $\Box$ și cu $\Diamond$ (teorema 17). De aici rezultă că în sistemul logicii temporale discutate, spre deosebire de **S4**, $\Box\Diamond\Box$ se va reduce la $\Diamond\Box$, iar $\Diamond\Box\Diamond$ la $\Box\Diamond$ (teoremele 18 și 19). Sistemul dispune de patru operatori distincți $\Box$, $\Diamond$, $\Box\Diamond$, $\Diamond\Box$ și considerînd și pe $\bigcirc$, avem următoarele patru tipuri:

$$\bigcirc^k\Box, \quad \bigcirc^k\Diamond, \quad \bigcirc^k\Box\Diamond \text{ și } \bigcirc^k\Diamond\Box.$$

Teoremele T20 și T21 sînt expresia proprietății de punct fix a operatorilor $\Box$ și $\Diamond$, în ecuațiile $X \subseteq p \wedge \bigcirc X$ și $X \equiv p \vee \bigcirc X$. Axioma A9 este o caracterizare analoagă pentru operatorul Δ și anume ca soluție a ecuației $X \equiv q \vee (p \wedge \bigcirc X)$.

Mai mult decît atît, să considerăm sistemul

$$X \to q \vee (p \wedge \bigcirc X)$$

$$X \to \Diamond q.$$

Axiomele A9 și A10 afirmă că $X = p \Delta q$ este o soluție a acestui sistem, iar regula R14 spune că orice soluție a acestui sistem trebuie să implice pe $p \Delta q$. Deci $p \Delta q$ este o soluție maximală a acestui sistem, $X \to p \Delta q$, $X \leqslant p \Delta q$.

Dacă se ține seama de regula R15, putem considera pe $p \Delta q$ ca o soluție minimală a ecuației

$$q \vee (p \wedge \bigcirc X) \to X$$

Scanned with OKEN Scanner

pentru că, conform acestei reguli, orice soluție a acestor ecuații este implicată de $p \Delta q : p \Delta q \to X$ și deci $X \leqslant p \Delta q$.

Dacă operatorul $\Box$ este un operator universal, iar $\Diamond$ unul existențial, $\bigcirc$ poate fi privit drept universal față de primul argument și existențial față de al doilea. Proprietățile din teoremele 29 și 30 sînt legate de acest aspect.

Singura posibilitate de absorbție a lui Δ este indicată de teoremele 37 și 38.

Operatorul binar ∇. Reamintim că $p \Delta q$ (p apoi q) implică „q va avea cîndva loc" (A10). Există o altă modalitate temporală care exprimă ideea „p dacă nu q". Operatorul acesta poate fi notat cu $p \nabla q$ și definit astfel :

$$p \nabla q \equiv \Box p \vee p \Delta q$$

Menționăm că alți autori folosesc un operator „p atîta timp cît q" care nu este altceva decît $q \nabla (\sim p)$.

Formula $p \nabla q = \Box \sim p \Delta q$ poate fi citită astfel : sau q după p, sau permanent p. Nu mai apare cerința ca q să se producă cîndva, ca în cazul operatorului Δ.

Regula care arată cum poate fi introdus acest operator în calcule este :

R19 Dacă $\vdash p \to \bigcirc(p \vee q)$, atunci $\vdash p \to u \nabla q$.

Legătura pe care o are operatorul Δ cu functorul modal S, introdus în 1942 la baza unei logici modale de către Gr. C. Moisil, confirmă alegerea făcută de logicianul român [4].

9.2. LOGICA DINAMICĂ PROPOZIȚIONALĂ

Dintre sutele de logici care au fost scrise în vederea aplicării lor la studiul programelor (corectitudine, terminare, echivalență etc.), citățm pe cea care este mai des folosită, logica dinamică propozițională PDL.

Logica dinamică are în alfabetul limbajului său două categorii de entități : a) $p, q, r, \ldots, P, Q, R, \ldots$ variabile propoziționale și formule și b) $a, b, c, \ldots, A, B, C, \ldots$ programe atomice și programe. Ideea fundamentală este de a considera programele drept modalități. Certul și posibilul modal, $\Box$ și $\Diamond$, se scriu deschis, cuprinzînd în centrul lor programul la care se referă. Astfel, citim :

$[a]p$ (programul) a se termină cu (aserțiunea) p

$\langle a \rangle p$ (programul) a se poate termina cu (aserțiunea) p.

Există atîtea modalități cîte programe sînt. Legătura dintre cele două notații [ ] și $\langle\ \rangle$ este analoagă cu cea din logica modală :

$$\langle a \rangle p = \sim [a] \sim p.$$

Programele sînt definite inductiv :

1) Orice program atomic $a, b, c, \ldots$ este program.

2) Dacă A și B sînt programe, tot așa sînt A^-, A^*, $(A \cup B)$, $(A ; B)$.

Operațiile introduse pentru programe sînt cele ce intervin în mînuirea programelor : A^* înseamnă „A se repetă de un număr nedeterminat de ori" ; $(A \cup B)$: „se ia A sau B nedeterminist" ; $(A ; B)$: „începe A, se termină B". În alte logici dinamice, se introduce $p?$ cu sensul de program test : încerci p și procedezi numai dacă p este adevărat.

Scanned with OKEN Scanner

Formulele se introduc la fel :

1) Orice variabilă propoziţională p, q, r, ... este formulă.

2) Dacă P, Q sînt formule, atunci $\neg P$, $P \to Q$ sînt formule; dacă A este program, $[A]P$ este formulă.

Sistemul formal PDL are următoarele axiome şi reguli :

1. $P \to (Q \to P)$
2. $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$
3. $(\sim P \to \sim Q) \to (Q \to P)$
4. $[A](P \to Q) \to ([A]P \to [A]Q)$
5. $[A \cup B]P \equiv [A]P \wedge [B]P$
6. $[A ; B]P \equiv [A] \cdot [B]P$
7. $[A^*]P \to [A]P$
8. $[A^*]P \to P$
9. $[A^*]P \to [A^*][A^*]P$
10. $P \to [A]\langle A^- \rangle P$
11. $P \to [A^-]\langle A \rangle P$
12. $(P \wedge [A^*](P \to [A]P)) \to [A^*]P$.

Regula MP : $\dfrac{P,\ P \to Q}{Q}$ şi $\dfrac{P}{[A]P}$

Prin definiţie : $P \vee Q = \sim P \to Q$, $P \wedge Q = \sim(P \to \sim Q)$, $(P \equiv Q) = (P \to Q) \wedge (Q \to P)$ şi $\langle A \rangle P = \sim[A] \sim P$.

Semantica acestui sistem foloseşte semantica relaţională a programelor, precum şi semantica relaţională din logica modală (structuri Kripke), necesare pentru demonstrarea compeltitudinii.

9.3. SISTEMUL R

Logica temporală bazată pe ideea de realizabilitate temporală se numeşte *sistemul* **R**. Operaţia $R_t(P)$ notează faptul că „formula P este realizată în momentul t" (este validă în momentul t). O propoziţie p definită temporal („la Bucureşti este cald la 1 iulie 1987") este echivalentă cu $R_t(p)$ pentru orice t, pentru că în orice moment o enunţăm aceasta nu îşi pierde valabilitatea. Iterarea operaţiei R_t ne conduce la regula

$$R_{t'}[R_t(p)] \equiv R_t(p).$$

Introducerea momentului privilegiat „acum" care se notează cu $t = n$ modifică această regulă sub forma

$$R_{t'}[R_t(p)] = \begin{cases} R_t(p), & t \neq n, \\ R_{t'}(p), & t = n, \end{cases}$$

pentru că o propoziție validă acum, realizată la momentul t', este validă la momentul t'. „Acum" mai are proprietatea :

$$R_n(p) \equiv p.$$

Construirea unei logici necesită următorul alfabet :

a) $p, q, r, \ldots$ variabile propoziționale (rezervăm ca de obicei majusculele pentru formule) ; b) variabilele t, t' pentru intervale de timp scurse de la origine (de exemplu *date*) ; c) simbolul n pentru „acum" ; d) conectorii calculului propozițional $\sim$, $\wedge$, $\rightarrow$, $\equiv$ (echivalența) ; e) R operatorul de realizare cronologică. În plus, trebuie să exprimăm : „toate momentele" prin cuantificatorul universal $(\forall t)$ și „există un moment" prin cuantificatorul existențial $(\exists t)$. Întrucît ne ocupăm de o logică propozițională, nu vom utiliza cuantificatorii decît pentru a exprima proprietăți ale variabilei t. De exemplu :

$(\exists t)(\forall t')(t' < t)$ Există un moment anterior t' oricărui moment t.

$(\forall t')\,(t_0 \leqslant t')$ Toate momentele sînt posterioare (sau coincid) cu originea t_0.

Următorul sistem de axiome caracterizează sistemul **R** :

R1 $R_t(\sim P) \equiv \sim R_t(P)$

R2 $R_t(P \wedge Q) \equiv R_t(P) \wedge R_t(Q)$

R3 $R_n(P) \equiv P$

R4 $R_{t'}((\forall t)P) \equiv (\forall t)R_{t'}(P)$

R5 $R_{t'}(R_t(P)) \equiv R_t(P)$

R6 $R_t(n = t') \equiv t = t'$

R7 $R_t(t' = t'') \equiv t' = t''$

R8 $(\forall t)P \rightarrow P^{t/n}$

Regulile de inferență sînt :

1) Cînd P este teoremă, la fel este $(\forall t)R_t(P)$.

2) Dacă $(\forall t)R_t(P)$ este teoremă, la fel este P (reciproca lui 1).

3) Cînd $\vdash P \equiv Q$, putem substitui pe P cu Q în orice teoremă.

R1 exprimă faptul că negația unei formule la un moment dat este realizată dacă și numai dacă formula nu este realizată la acel moment. R2 afirmă că realizarea conjuncției este conjuncția realizărilor. R3 și R5 au fost discutate mai sus. R6 și R7 justifică utilizarea egalității $=$ pentru momentele de timp. R8 arată că dacă o formulă este realizată în toate momentele, putem face substituția t/n, pentru orice apariție liberă a lui t (care nu cade sub incidența vreunui cuantificator în P).

Prin dotarea cu o semantică de tipul celei descrise la logicile modale (v. 8.8) se demonstrează completitudinea lui **R**.

9.4. LOGICA TIMPURILOR

Logica temporală nu are trecut, prezent și viitor. *Logica timpurilor* trece de la un „timp generic" la un „timp specific". În acest scop este folosită relația de precedență Utt' (t este anterior lui t'). Întrucît prezentul

Scanned with OKEN Scanner

este notat cu n, atunci Utn reprezintă momentele trecute și Unt pe cele viitoare.

Se introduc următorii operatori modali, în dreptul cărora se arată interpretarea și definiția:

Gp	p întotdeauna în viitor	$(\forall t)[Unt \to R_t(p)]$
Hp	p întotdeauna în trecut	$(\forall t)\,[Utn \to R_t(p)]$
Fp	p cîndva în viitor	$(\exists t)\,[Unt \wedge R_t(p)]$
Pp	p cîndva în trecut	$(\exists t)\,[Utn \wedge R_t(p)]$

Sistemul minimal $\mathbf{K_t}$ al logicii timpurilor cuprinde următoarele axiome:

G1 $G(p \to q) \to (Gp \to Gq)$

H1 $H(p \to q) \to (Hp \to Hq)$

G2 $\sim H \sim Gp \to p$

H2 $\sim G \sim Hp \to p$

și reguli:

1. Dacă formula Q este o tautologie, $\vdash Q$.
2. Dacă $\vdash Q$, atunci $\vdash HQ$.
3. Dacă $\vdash Q$, atunci $\vdash GQ$.
4. Dacă $\vdash Q$ și $\vdash Q \to R$, atunci $\vdash R$. (MP)

Sistemul $\mathbf{K_b}$ se obține din $\mathbf{K_t}$ prin adăugarea următoarelor trei axiome:

G3 $Gp \to GGp$

H3 $Hp \to HHp$

H4 $(H(p \vee q) \wedge H(p \vee Hq) \wedge H(Hp \vee q)) \to (Hp \vee Hq)$

regulile de inferență rămînînd aceleași ca în $\mathbf{K_t}$.

Sensul acestor axiome este că ele exprimă (G3 și H3) o proprietate de tranzitivitate a relației U:

$$(\forall t_1)(\forall t_2)(\forall t_3)\,[Ut_1t_2 \wedge Ut_2t_3 \to Ut_1t_3]$$

și o proprietate de „liniaritate înapoi":

$$(\forall t_1)(\forall t_2)(\forall t_3)\,[Ut_1t_3 \wedge Ut_2t_3) \to (Ut_1t_2 \vee t_1 = t_2 \vee Ut_2t_1)].$$

Structura timpului este aceea de arbore: înapoi, dintr-un punct se ajunge numai într-un singur punct, dar înainte există oricînd posibilitatea unei ramificații (numele sistemului $\mathbf{K_b}$ vine de la „branching").

Iată o listă de teoreme în $\mathbf{K_b}$ (vom demonstra numai T5', lăsînd celelalte demonstrații ca exerciții):

T1' $p \to G \sim H \sim p$

T2' $Gp \wedge Fq \to F(p \wedge q)$

T3' $F(p \vee q) \to Fp \vee Fq$

T4' Dacă $\vdash A \to B$, atunci $\vdash FA \to FB$

T5' $F(Pp \wedge Pq) \to [FP(p \wedge q) \vee FP(p \wedge Pq) \vee FP(P_p \wedge p)]$

Scanned with OKEN Scanner

Demonstrație.

a) $Pp \wedge Pq \to P(p \wedge q) \vee P(p \wedge Pq) \vee P(Pp \wedge q)$ Din H4

b) $F(Pp \wedge Pq) \to F[(P(p \wedge q) \vee P(p \wedge Pq) \vee P(Pp \wedge q)]$ Din a) și T4′

c) $F(Pp \wedge Pq) \to [FP(p \wedge q) \vee FP(p \wedge Pq) \vee FP(Pp \wedge q)]$ Din T3′

T6′ $\sim FP(\sim p \wedge Gp \wedge Hp \wedge p)$

T7′ $\sim FP(P \sim p \wedge Gp \wedge Hp \wedge p)$

T8′ $\sim FP(\sim p \wedge P(Gp \wedge Hp \wedge p))$

T9′ $(Gp \wedge Hp \wedge p) \to GHp.$

Sistemele $\mathbf{K_l}$, $\mathbf{K_l^{+\infty}}$, $\mathbf{K_l^{-\infty}}$, $\mathbf{K_l^{\pm\infty}}$. Sistemul $\mathbf{K_b}$ nu este simetric, neexistînd o axiomă pentru G similară cu H4. Dacă o introducem, obținem un sistem $\mathbf{K_l}$ complet liniar (înainte și înapoi) format din axiomele G1 — G4, H1 — H4, unde

G4 $[G(p \vee q) \wedge G(p \vee Gp) \wedge G(Gp \vee q)] \to Gp \vee Gq.$

Diferitele sisteme, ca de exemplu $\mathbf{K_b}$ și $\mathbf{K_l}$, se obțin din sistemul minimal $\mathbf{K_t}$ prin adăugarea unor axiome noi care reflectă ipotezele asupra timpului.

Există o axiomă „foarfece" care arată că timpul are sfîrșit. Aceasta este $Gp \vee FGp$. Dacă sîntem în ultimul moment t, atunci Gp este adevărată pentru că $t < t' \to R_{t'}(p)$ este adevărată, antecedentul fiind fals, iar dacă nu sîntem în ultimul moment, FGp este adevărată. Axioma $Hp \vee \vee PHp$ indică în mod analog, că timpul are un prim moment (un început).

Dacă vrem să spunem că timpul viitor nu are sfîrșit, ne servim de axioma $Gp \to Fp$. În ipoteza adoptată, dacă Gp este adevărată la momentul t, există un moment t', $t < t'$, în care p este adevărată, deci Fp este adevărată în momentul t.

Analog axioma $Hp \to Pp$ enunță faptul că timpul trecut nu are sfîrșit. Numim aceste două axiome

G5 $Gp \to Fp$, H5 $Hp \to Pp.$

Dacă adăugăm pe G5 la sistemul $\mathbf{K_l}$, obținem sistemul $\mathbf{K_l^{+\infty}}$ (timp viitor infinit). Dacă adăugăm H5 la sistemul $\mathbf{K_l}$, obținem sistemul $\mathbf{K_l^{-\infty}}$ (timp trecut nesfîrșit). Dacă adăugăm G5 și H5 la axiomele lui K_l, definim sistemul $\mathbf{K_l^{\pm\infty}}$ (timp infinit trecut și viitor).

Alte axiome sînt legate de noi ipoteze asupra timpului:

G6 $GGp \to Gp$ timp dens

H6 $HHp \to Hp$

G7 $\Box(Gp \to p) \to (Gp \to Hp)$ timp discret

G8 $\Box(Gp \to PGp) \to (Gp \to Hp)$ timp continuu.

Simbolul modal $\Box$ a fost luat aici în sensul „întotdeauna adevărat":

$$\Box Q \equiv GQ \wedge HQ \wedge Q.$$

Completitudinea sistemelor $\mathbf{K_t}$, $\mathbf{K_b}$, $\mathbf{K_l}$ se demonstrează prin folosirea modelelor lui Kripke.

Scanned with OKEN Scanner

9.5. LOGICA TIMPURILOR ȘI LOGICILE MODALE

Există două maniere de a traduce în operatorii clasici modali ($\Box$ necesitatea și $\Diamond$ posibilul) operatorii GH, F, P după cum în definirea necesității (sau posibilului) nu luăm sau luăm în considerație trecutul.

Conform primei interpretări, numite *diodoriană*:

$$(\mathrm{D}\Box) \qquad \Box Q \equiv Q \wedge GQ$$

$$(\mathrm{D}\Diamond) \qquad \Diamond Q \equiv Q \wedge FQ$$

iar conform celei de-a doua, numită *aristotelică*:

$$(\mathrm{A}\Box) \qquad \Box Q \equiv Q \wedge GQ \wedge HQ$$

$$(\mathrm{A}\Diamond) \qquad \Diamond Q \equiv Q \wedge FQ \wedge PQ.$$

Se demonstrează că din adăugarea lui $\mathrm{D}\Box$ la $\mathbf{K}_t$ rezultă sistemul modal **T**. Din $\mathrm{A}\Box$ și $\mathbf{K}_t$ rezultă sistemul **B**; deci toate teoremele lui **B** sînt demonstrabile în $\mathbf{K}_t$ plus $\mathrm{A}\Box$.

Din sistemul $\mathbf{K}_b$ la care se adaugă $\mathrm{D}\Box$ rezultă sistemul cunoscut **S4**, dar ce rezultă prin suplimentarea lui K_b cu $A\Box$ este o chestiune încă deschisă.

Sistemul $\mathbf{K}_l$ (nediscret) plus $\mathrm{D}\Box$ ne conduce la sistemul **S4.3** iar sistemul $\mathbf{K}_l$ discret conduce la sistemul modal **D**. Din sistemul $\mathbf{K}_l$ liniar cu $\mathrm{A}\Box$ rezultă sistemul modal **S5**. (Sistemele modale citate au fost definite în 8.5.)

Rezumăm definițiile și proprietățile logicilor timpurilor prezentate prin tabelul 9.1.

Tabelul 9.1

Sisteme	Axiome + 4 reguli	Sisteme modale obținute prin adăugarea condițiilor	
		$\mathrm{D}\Box$	$\mathrm{A}\Box$
$\mathbf{K}_t$	G1, G2, H1, H2	**T**	**B**
$\mathbf{K}_b$	G1–G3, H1–H4	**S4**	?
$\mathbf{K}_l$	G1–G4, H1–H4	**S4.3**	**S5**
$\mathbf{K}_l$ (Timp discret)	+ G7	**D**	**S5**
$\mathbf{K}_l^{+\infty}$	G1–G5, H1–H4		
$\mathbf{K}_l^{-\infty}$	G1–G4, H1–H5		
$\mathbf{K}_l^{\pm\infty}$	G1–G5, H1–H5		

9.6. NOTĂ ISTORICĂ

Introducerea explicită a timpului în definiția modalității logice este legată de numele lui Diodor Cronos, din școala megarică. În spiritul paradoxurilor discutate de megarici sub denumirile de *argumente* (ca de pildă argumentul mincinosului: „minte cineva cînd nu minte?"), Diodor Cronos a elaborat *argumentul dominator*, despre care se publică studii și astăzi. Față de concepția megarică a posibilului în care trecutul și prezentul erau elementele principale, Diodor Cronos definește posibilul ca ceea ce este

Scanned with OKEN Scanner

acum sau poate fi în viitor : $\Diamond p = p \vee Fp$. Boeţiu scrie clar : „Diodor defineşte posibilul ca ceea ce este sau va fi (*quod aut est aut erit*)". Diodor foloseşte conceptul de timp şi în discutarea implicaţiei, unde s-a ridicat împotriva definiţiei filoniene (implicaţia materială). Astfel „Dacă soarele a răsărit, este ziuă" se poate enunţa temporal : „orice timp cînd soarele a răsărit este un timp cînd este ziuă". Putem data vremea cînd a trăit Diodor, după o legendă lansată de duşmanii săi atenieni. Diodor s-ar fi sinucis cînd n-a putut rezolva pe loc o problemă de logică ce i s-a pus în prezenţa lui Ptolemaios Soter, care cucerise Megara la 307 î.e.n.

Regăsim modalităţile temporale la Avicenna şi la logicienii arabi (care au tratat nu mai puţin de 13 modalităţi) şi în Europa medievală. La Pseudo-Scotus întîlnim patru feluri de necesitate temporalizată (*quando, pro semper, conditionale, ut nunc*). La mulţi autori medievali domină interpretarea dată de Aristotel, conform căruia necesarul este ceea ce este adevărat întotdeauna (nu numai în prezent şi în viitor).

Studiile de logică temporală au fost reanimate după cel de-al doilea război mondial prin contribuţiile lui Jerzy Los, Arthur Prior, Nicholas Rescher şi alţii. Lui Los i se datorează axiomatizări de tipul R (1948). Logica minimală a timpurilor a fost elaborată de Jack Lemmon (în anii '60). Un interes pentru punctul de vedere temporal l-a produs logica schimbării a lui G. H. von Wright (1963—1965).

Din anii '70 perspectiva asupra logicilor temporale s-a schimbat mult. Abordarea nu este atît filozofică, cît practică, determinată de necesitatea elaborării de logici adecvate pentru descrierea comportării programelor de calculator în vederea verificării automate a corectitudinii, terminării şi echivalenţei lor. Cu aceasta s-a intrat în etapa logicilor programelor şi a logicilor dinamice, care caracterizează preocupările intense din prezent.

9.7. EXERCIŢII ŞI PROBLEME

1. Să se demonstreze teoremele T3 — T11, T13 — T17, T19, T21 — T22, T25 — T26, T28, T30, T33 — T34, T36, T38 (v. 9.1).

2. Să se demonstreze în PDL (v. 9.2) formulele

a) $\langle A ; B\rangle P = \langle A\rangle \langle B\rangle P$,

b) $\langle A \vee B\rangle P = \langle A\rangle P \vee \langle B\rangle P$,

c) $\langle A\rangle (P \vee Q) = \langle A\rangle P \vee \langle A\rangle Q$.

3. Să se identifice eroarea în următoarea deducţie în R (v. 9.3):

a) $(\forall t)[R_t(p) \equiv R_t(p)]$ prin regula 1,

b) $R_t(p) \equiv R_t[R_t(p)]$ R5,

c) $(\forall t)\,[R_t(R_t(p) \equiv R_t(p))]$ b) şi regula 1,

d) $(\forall t)\, R_t[R_t(p) \equiv p]$ c), R1, R2, regula 3,

e) $R_t(p) \equiv p$ d), regula 2.

4. Să se arate că în sistemul R (v.9.3) putem înlocui pe R2 cu

$$R2' \quad R_t(P) \wedge R_t(Q) \rightarrow R_t(P \wedge Q).$$

Scanned with OKEN Scanner

5. Să se arate că sistemul care are aceleași axiome ca și R (v. 9.3), dar în care R5 este înlocuită cu

$$R5' \quad R_t'[R_t(p)] \rightarrow R_t(p),$$

este echivalent cu R.

6. Să se demonstreze că formulele

a) $[(\sim p \vee \sim Gq \vee r) \vee G(\sim s \vee \sim q \vee u)] \rightarrow [(\sim p \vee \sim Gq \vee r) \vee G(\sim s \vee u)]$,

b) $[H(\sim p \vee \sim Gq \vee r) \vee (\sim s \vee \sim q \vee t)] \rightarrow [H(\sim p \vee \sim Gq \vee r) \vee (\sim s \vee t)]$

sînt teoreme în $\mathbf{K}_t$ (v. 9.4).

7. Să se demonstreze teoremele T3' — T4', T6' — T9' de la 9.4.

8. Să se arate că axiomele (T)

$$\Box p \rightarrow p$$

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$$

sînt teoreme în sistemul $\mathbf{K}_t + D\Box$ (v. 9.5).

9. Să se arate că axioma (B)

$$\Diamond\Box p \rightarrow p$$

este o teoremă în sistemul $\mathbf{K}_t + A\Box$ (v. 9.5).

10. Să se arate că axioma (S4)

$$\Box p \rightarrow \Box\Box p$$

este o teoremă în sistemul $\mathbf{K}_t + D\Box$ (v. 9.5).

BIBLIOGRAFIE

1. Hoare, C.A.R. *An Axiomatic Basis for Computer Programming.* În : C.A.C.M., 12, p. 576—580, 1969.
2. Fischer, M., Ladner, R. *Propositional Dynamic Logic of Regular Programs.* În : Journal of Computer and System Sciences, Vol. XVIII, Nr. 2, Aprilie, 1979.
3. Manna, Z., Pnueli, A. *Verification of Concurrent Programs : a Temporal Proof System.* În : Foundations of Computer Science, IV. Math. Centrum, Amsterdam, 1983.
4. Malița Mihaela, Malița, M. *Functorul lui Moisil și logicile temporale.* În : Noesis, 1987.
5. Parikh, R. *Propositional Dynamic Logics of Programs : A Survey. Logic of Programs.* Berlin, Heidelberg, New York, Ed. Erwin Engeler, Springer Verlag, 1981.
6. Pratt, V.R. *Dynamic Logic.* În : Proc. 6-th International Congress for Logic, Philosophy and Methodology of Science, Hanover, Aug., 1979.
7. Prior, A. N., *Time and Modality.* Oxford at the Clarendon Press, 1957.
8. Prior, A. N. *Tense-Logic and the Continuity of Time.* În : Studia Logica, 13, 1962.
9. Prior, A. N. *Past, Present and Future.* Oxford at the Clarendon Press, 1967.
10. Radev, S. *Extension of P.D.L. and Consequence Relations.* Computation Theory. În : Fifth Symposium, Zaborow, Poland, Springer Verlag, 1985.
11. Rescher, N., Urquhart, A. *Temporal Logic.* New York, Springer Verlag, 1971.
12. Valiev, B. *On Axiomatisation of Deterministic Propositional Dynamic Logic.* În : Symposium on Mathematical Foundations of Computer Science. Olomano. Czechoslovakia, Sept., 1979.
13. Von Wright, G. H. *And Next.* În : Acta Philosophica, Fennica, 18, 1965.

Scanned with OKEN Scanner

10.

ELEMENTE DE LOGICĂ PROBABILISTĂ

10.1. PROBABILITATEA CA MĂSURĂ A REALIZABILITĂȚII FORMULELOR

Logica polivalentă și logica modală reprezintă tentativa de modelare a raționamentului asupra incertitudinii, fie utilizînd valori intermediare de adevăr (posibilul), fie introducînd modalitățile : necesar (cert) $\Box$ și posibil $\Diamond$. O alternativă este încorporarea conceptului de certitudine în calculul propozițiilor prin atașarea la fiecare formulă a unei măsuri a gradului ei de certitudine.

Din numeroasele căi de a introduce o măsură probabilistă a enunțurilor (formulelor) vom opta pentru cea care are un avantaj intuitiv și anume prin asimilarea inițială a gradului de certitudine a formulelor cu măsura realizabilității lor.

În calculul propozițional clasic (v. cap. 1) atenția este îndreptată spre două clase de formule : tautologiile (formulele adecvate în toate interpretările) și contradicțiile (formulele nerealizabile în nici o interpretare). Interesul pentru aceste două clase de formule este justificat de rolul pe care ambele îl joacă în sistemele deductive (v. cap. 2). În unele aplicații ale CP cum ar fi circuitele cu contacte (v. cap. 5), tautologiile și contradicțiile își pierd importanța, întregul interes fiind îndreptat spre formulele realizabile (consistente), singurele apte de a reprezenta circuite utile.

Întrebarea elementară, rămasă fără răpuns în CP, este : cît de realizabilă este o formulă realizabilă? Răspunsul, tot elementar, constă în încercarea de introducere a unei *măsuri a realizabilității*. Cerințele unei asemenea măsuri rezultă în mod natural din proprietățile formulelor realizabile (ne situăm în ipoteza unui număr finit de variabile propoziționale). Fie $R : F \to [0, 1]$ o aplicație a mulțimii formulelor F pe mulțimea numerelor reale cuprinse între 0 și 1, care satisface următoarele condiții :

1. $R(A) \geqslant 0$.

2. $\models A \Leftrightarrow B \Rightarrow R(A) = R(B)$ (două formule echivalente au aceeași măsură de realizabilitate).

3. $\models A$ dacă și numai dacă $R(A) = 1$ (unei tautologii i se atribuie măsura maximă de realizabilitate).

4. Dacă $\models \overline{AB}$, adică dacă A și B sînt incompatibile (nu pot fi adevărate în același timp), atunci

$$R(A \vee B) = R(A) + R(B).$$

Scanned with OKEN Scanner

Această condiție exprimă considerația intuitivă că $A \vee B$ se realizează ori de cîte ori se realizează A și ori de cîte ori se realizează B, cele două formule nerealizîndu-se niciodată simultan.

Pentru facilitarea scrierii formulelor scriem AB pentru $A \wedge B$.

Consecința imediată a lui 3 și 4 este

5. $R(A) + R(\bar{A}) = 1,$

de unde rezultă că A este o contradicție, dacă și numai dacă $R(A) = 0$.

6. $R(AB) = R(A) + R(B) - R(A \vee B)$, oricare ar fi formulele A, B.

Obținem pe 6, aplicînd pe 2 și 4 formulelor $A \vee B \Leftrightarrow A \vee \bar{A}B$ și $B \Leftrightarrow AB \vee \bar{A}B$, unde formulele $A \vee \bar{A}B$ și $AB \vee \bar{A}B$ îndeplinesc condițiile cerute de 4, și din $R(A \vee B) = R(A) + R(\bar{A}B)$ și $R(B) = R(AB) + R(\bar{A}B)$.

Întrucît orice formulă din CP se poate scrie sub formă normală disjunctivă

$$A = \bigvee_{j=1}^{2^n} d_j m_j,$$

unde d_j sînt 1 sau 0, iar m_j minitermeni (constituenți sau clauze conjunctive), obținem

7. $R(A) = \sum_{j=1}^{2^n} d_j R(m_j),$

deoarece $m_j m_k = 0$ $(j \neq k)$.

În consecință, dacă A este tautologie, $\sum_{j=1}^{2^n} R(m_j) = 1$.

Din 7 rezultă că măsura de realizabilitate a unei formule oarecare este unic determinată de cunoașterea realizabilității minitermenilor.

Proprietatea de bază a minitermenilor este aceea de a fi adevărați într-o singură interpretare și de a fi falsificați în toate celelalte. Vom numi *lumi realizabile* L_j cele 2^n interpretări posibile. Definim realizabilitatea unei lumi drept realizabilitatea formulei care este adevărată numai în această lume.

8. $R(L_j) = R(m_j)$; $\quad j = 1, 2, \ldots, 2^n$.

Dar $d_j = 1$, cînd A este adevărată în lumea L_j și $d_j = 0$ cînd A este falsă în lumea L_j. Obținem astfel următoarea interpretare semantică a măsurii realizabilității unei formule A : *ea este suma realizabilității lumilor în care A este adevărată.*

Formulînd ipoteza că în CP lumile sînt egal realizabile (considerînd sistemul logic cu maximă entropie), putem conveni următoarea măsură a realizabilității lumilor (sau minitermenilor) :

9. $R(L_j) = R(m_j) = \frac{1}{2^n}, \quad j = 1, \ldots, 2^n,$

unde se ține seama de consecința lui 7. Ca și în relația 7, numărul n este numărul variabilelor propoziționale ale formulei A.

Un minitermen de n variabile se reprezintă sub formă de conjuncții de literali (variabile complementate sau necomplementate) $l_1 \ldots l_n$, iar

Scanned with OKEN Scanner

Această condiție exprimă considerația intuitivă că $A \vee B$ se realizează ori de cîte ori se realizează A și ori de cîte ori se realizează B, cele două formule nerealizîndu-se niciodată simultan.

Pentru facilitarea scrierii formulelor scriem AB pentru $A \wedge B$.

Consecința imediată a lui 3 și 4 este

5. $R(A) + R(\bar{A}) = 1,$

de unde rezultă că A este o contradicție, dacă și numai dacă $R(A) = 0$.

6. $R(AB) = R(A) + R(B) - R(A \vee B)$, oricare ar fi formulele A, B.

Obținem pe 6, aplicînd pe 2 și 4 formulelor $A \vee B \Leftrightarrow A \vee \bar{A}B$ și $B \Leftrightarrow AB \vee \bar{A}B$, unde formulele $A \vee \bar{A}B$ și $AB \vee \bar{A}B$ îndeplinesc condițiile cerute de 4, și din $R(A \vee B) = R(A) + R(\bar{A}B)$ și $R(B) = R(AB) + R(\bar{A}B)$.

Întrucît orice formulă din CP se poate scrie sub formă normală disjunctivă

$$A = \bigvee_{j=1}^{2^n} d_j m_j,$$

unde d_j sînt 1 sau 0, iar m_j minitermeni (constituenți sau clauze conjunctive), obținem

7. $R(A) = \sum_{j=1}^{2^n} d_j R(m_j),$

deoarece $m_j m_k = 0$ $(j \neq k)$.

În consecință, dacă A este tautologie, $\sum_{j=1}^{2^n} R(m_j) = 1$.

Din 7 rezultă că măsura de realizabilitate a unei formule oarecare este unic determinată de cunoașterea realizabilității minitermenilor.

Proprietatea de bază a minitermenilor este aceea de a fi adevărați într-o singură interpretare și de a fi falsificați în toate celelalte. Vom numi *lumi realizabile* L_j cele 2^n interpretări posibile. Definim realizabilitatea unei lumi drept realizabilitatea formulei care este adevărată numai în această lume.

8. $R(L_j) = R(m_j)\,; \qquad j = 1, 2, \ldots, 2^n.$

Dar $d_j = 1$, cînd A este adevărată în lumea L_j și $d_j = 0$ cînd A este falsă în lumea L_j. Obținem astfel următoarea interpretare semantică a măsurii realizabilității unei formule A : *ea este suma realizabilităților lumilor în care A este adevărată.*

Formulînd ipoteza că în CP lumile sînt egal realizabile (considerînd sistemul logic cu maximă entropie), putem conveni următoarea măsură a realizabilității lumilor (sau minitermenilor) :

9. $R(L_j) = R(m_j) = \frac{1}{2^n}, \qquad j = 1, \ldots, 2^n,$

unde se ține seama de consecința lui 7. Ca și în relația 7, numărul n este numărul variabilelor propoziționale ale formulei A.

Un minitermen de n variabile se reprezintă sub formă de conjuncții de literali (variabile complementate sau necomplementate) $l_1 \ldots l_n$, iar

Scanned with OKEN Scanner

un minitermen de $n-1$ variabile poate fi scris

$$l_1 \dots l_{n-1} = l_1 \dots l_{n-1} l_n + l_1 \dots l_{n-1} \bar{l}_n.$$

Aplicînd formulele 4 și 9, obținem :

10. $R(l_1 \dots l_n) = \frac{1}{2^n}$, $R(l_1 \dots l_{n-1}) = \frac{1}{2^{n-1}}$, $R(l) = \frac{1}{2}$.

Măsura realizabilității unei formule poate astfel să fie interpretată drept numărul coeficienților d_j pozitivi, adică numărul lumilor posibile în care formula se realizează (este adevărată) împărțit la numărul total al lumilor posibile.

Exemplu. Fie $A = x_1x_2x_3 + x_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3 = \sum_{j=1}^{2^3} d_j m_j$. În această formulă $d_1 = d_2 = d_4 = d_7 = d_8 = 1$ și $d_3 = d_5 = d_6 = 0$ (ordonarea minitermenilor s-a făcut după criteriul tablei de adevăr utilizate în cap. 1).

Aplicăm formula $R(A) = \sum_{j=1}^{8} d_j R(m_j) = 5 \cdot \frac{1}{8} = \frac{5}{8}$.

Simplificînd formula A, obținem $A' = x_1x_2 + \bar{x}_1\bar{x}_2 + \bar{x}_1\bar{x}_3$. Întrucît $A \leftrightarrow A'$, $R(A') = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{8} = \frac{5}{8}$.

Aplicație. Măsura realizabilității poate fi utilizată pentru demonstrarea caracterului de tautologie sau nerealizabilitate a unei formule date.

a) Fie $A = (P \to Q)(Q \to R) \to (P \to R)$. Se observă că A este tautologie pentru că $A = P\bar{Q} + Q\bar{R} + \bar{P} + R$ și

$$R(A) = \frac{1}{4} + \frac{1}{4} + \frac{1}{2} + \frac{1}{2} - \frac{1}{8} - \frac{1}{8} - \frac{1}{4} = 1.$$

b) $A = (P \vee Q)(Q \vee R)(R \vee W)(\bar{R} \vee \bar{P})(\bar{W} \vee \bar{Q})(\bar{Q} \vee \bar{R})$ este nerealizabilă. Cercetăm dualul

$$B = PQ + QR + RW + \bar{R}\bar{P} + \bar{W}\bar{Q} + \bar{Q}\bar{R},$$

iar $R(B) = 6 \cdot \frac{1}{4} - \frac{1}{8} - \frac{1}{16} - \frac{1}{8} - \frac{1}{8} + \frac{1}{16} + \frac{1}{16} - \frac{1}{8} - \frac{1}{16} = 1$

Comparăm proprietățile lui R cu axiomele probabilității în enunțul lui Kolmogorov, pentru cazul finit :

P1 $p(A) \geqslant 0$,

P2 dacă $\models A$, atunci $p(A) = 1$,

P3 dacă $\models A \to B$, atunci $p(A) \leqslant p(B)$,

P4 dacă $\models \overline{A\,B}$, atunci $p(A \vee B) = p(A) + p(B)$.

Funcția R satisface axiomele P1, P2, P4 conform proprietăților 1, 3, 4. În ceea ce privește P3, este și ea satisfăcută :

$$1 = R(A \to B) = R(\bar{A}) + R(B) - R(\bar{A}B) = 1 - R(A) + R(B) - R(\bar{A}B),$$

de unde $R(A) \leqslant R(B)$ (am folosit formula $A \to B \Leftrightarrow \bar{A} \vee B$). Proprietatea P3 este importantă, întrucît arată că într-o implicație strictă (implicație tautologică) sau într-o relație de consecință logică, probabilitatea antecedentului este mai mică sau egală cu probabilitatea consecventului.

Scanned with OKEN Scanner

10.2. CALCULUL PROPOZIȚIILOR ȘI CALCULUL PROBABILITĂȚILOR

Conceptul de realizabilitate se dovedește una din căile cele mai naturale pentru introducerea unei logici probabiliste. Avantajul este dobîndirea ideii obiective de frecvență a adevărului, măsurată de probabilitatea asociată fiecărei formule : în cîte lumi formula este adevărată din totalul lumilor posibile. De acum înainte vom scrie $p(A)$ în loc de $R(A)$ înțelegînd prin p măsura introdusă prin P1 — P4 pentru formulele unui limbaj de ordinul zero. Convenim să vedem în $p(A)$ o măsură a gradului de certitudine a formulei A. Gradul de certitudine a tautologiilor este 1 și al contradicțiilor 0. Gradul de certitudine a formulelor realizabile este diferit de 0. Ne situăm în continuare în cazul finit, dar părăsim ipoteza echiprobabilității lumilor posibile.

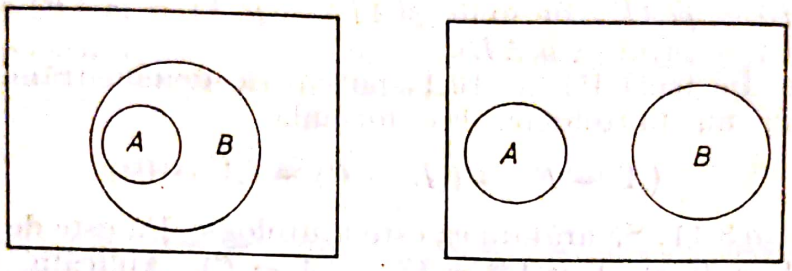


Fig. 10.1

Recurgînd la reprezentarea grafică cunoscută a probabilităților sub formă de arii, vom avea o imagine a axiomelor A3 și A4 în fig. 10.1.

Relația de consecință logică $A \Rightarrow B$ sau $\models A \to B$ se reprezintă prin $A \subseteq B$, de unde $p(A) \leqslant p(B)$. Cazul incompatibilității este reprezentat de cercurile disjuncte, a căror sumă de arii reprezentă probabilitatea ca A sau B să fie adevărate.

Presupunem că relația de ordine din P3 între gradele de certitudine este o relație de ordine liniară (care satisface condițiile : 1) $x \leqslant x$; 2) din $x \leqslant y$ și $y \leqslant z$ rezultă $x \leqslant z$; 3) $x \leqslant z$ sau $z \leqslant x$; 4) din $x \leqslant y$ și $y \leqslant x$ rezultă $x = y$).

Mai presupunem că dacă înlocuim formulele unei disjuncții cu formule cu o probabilitate mai mică sau egală, probabilitatea disjuncției ce rezultă va fi mai mică sau egală cu probabilitatea disjuncției inițiale, dacă substituția s-a făcut cu formulele incompatibile. Cu alte cuvinte, din $p(A) \leqslant p(B)$ și $p(C) \leqslant p(D)$ rezultă $p(A \vee C) \leqslant p(B \vee D)$ cu condiția $\models \overline{BD}$.

Teoremele care se deduc din P1 — P4 formează baza utilizării calculului probabilităților într-un calcul propozițional :

T1 $p(\bar{A}) = 1 - p(A)$. Analog cu 5 din 10.1.

T2 $p(A \vee B) = p(A) + p(B) - p(AB)$. Analog cu 6 din 10.1.

T3 $p(A \vee B) \leqslant p(A) + p(B)$. Rezultă din T2 și P1.

T4 Dacă $\models A \leftrightarrow B$, atunci $p(A) = p(B)$. Rezultă din aplicarea P3 la dubla implicație. Analog cu 3 din 10.1.

T5 $p(A) \leqslant p(A \vee B)$. Rezultă din P3.

T6 $p(AB) \leqslant p(A)$. La fel,

T7 $p(AB) \geqslant p(A) + p(B) - 1$. Rezultă din P1 și T2.

T8 Dacă $\models AB \to C$, atunci $p(A) + p(B) - 1 \leqslant p(C)$. Din T7 și P3.

Scanned with OKEN Scanner

T9 Dacă $A \leqslant \bigvee_{1}^{2^n} m_j d_j$, atunci $p(A) = \sum_{i=1}^{2^k} p(m_i) d_i$.

Ținînd seama că m_i sînt conjuncții incompatibile două cîte două, $m_i m_j = 0$ $(i \neq j)$, condițiile aplicării lui P4 sînt întrunite. Analog cu 7 din 10.1.

T10 $p(A \to B) + p(A) - 1 \leqslant p(B) \leqslant p(A \to B)$. Scriem $p(A \to B) = p(\bar{A} \vee B)$ și avem

T11 $p(\bar{A}) = 1 - p(A) \leqslant p(\bar{A} \vee B) \leqslant p(\bar{A}) + p(B) = 1 - p(A) + p(B)$ și deci $p(A \to B) + p(A) - 1 \leqslant p(B)$. Pe de altă parte $p(B) \leqslant p(A \to B)$.

T12 $p(A \to B) = 1 - p(A) + p(AB)$.

Avem $A \to B \Leftrightarrow \bar{A} \vee B \Leftrightarrow \overline{A\bar{B}}$, deci $p(A \to B) = 1 - p(A\bar{B})$. Dar $p(A) = p(AB) + p(A\bar{B})$, de unde $p(A\bar{B}) = p(A) - p(AB)$ și deci $p(A \to B) = 1 - p(A) + p(AB)$.

Aplicație. Pe baza P2 (v. 10.1) putem să demonstrăm că o formulă dată este sau nu tautologie. Fie formula

$$(A \to B) \to ((B \to C) \to (A \to C))$$

care este T1 din 3.11. Să arătăm că este tautologie. Ea este de forma $X \to Y$, unde $X \leqslant A \to B$ și $Y = (B \to C) \to (A \to C)$. Aplicăm formula T12 din 10.2:

$$p(X \to Y) = p(XY) - p(X) + 1$$

$$X\bar{Y} = (\bar{A} + B)((\bar{B} + C) \to (\bar{A} + C)) = (\bar{A} + B)(B\bar{C} + \bar{A} + C) =$$

$$= (\bar{A} + B)(\bar{A} + B + C) = \bar{A} + B = X.$$

Deci

$$p((A \to B) \to ((B \to C) \to (A \to C))) = 1.$$

Criteriul probabilist corespunde următorului criteriu algebric: $X \to Y$ este tautologie cînd $X \Rightarrow Y$, deci cînd $X \leqslant Y$ sau $XY = X$.

10.3. PROBLEMA LUI BOOLE

Vom ilustra problema printr-un exemplu simplu. Fie $g(x, y) = x + y$ o funcție booleană. (Vom utiliza notația folosită la polinoamele booleene, numind formulele propoziționale funcții). Să presupunem că sînt date probabilitățile lui x și y. Între ce margini (superioară și inferioară) trebuie să se încadreze probabilitatea disjuncției $p(g) = p(x + y)$? Dacă ținem seama că probabilitatea unei formule este suma probabilităților lumilor în care este adevărată, putem concentra în formă matriceală ceea ce știm despre x, y, $x + y$:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} p_1 + p_2 \\ p_1 + p_3 \\ p_1 + p_2 + p_3 \end{bmatrix} = \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix}$$

unde Π_1, Π_2 sînt probabilitățile (date) ale lui x și y; Π_3 este probabilitatea (asupra căreia trebuie să ne pronunțăm) a lui $x + y$: p_1, p_2, p_3, p_4 sînt probabilitățile celor 2^2 lumi posibile ce corespund la 2 variabile propoziționale x și y, iar liniile primei matrice sînt valorile de adevăr ale lui x, y și $x + y$ în cele 4 lumi.

Scanned with OKEN Scanner

Se observă că $\Pi_1 \leqslant \Pi_3 \leqslant \Pi_1 + \Pi_2$ pentru că $\Pi_3 = p_1 + p_2 + p_3$, $\Pi_1 = p_1 + p_2$ $\Pi_2 = p_1 + p_3$ și că $\Pi_2 \leqslant \Pi_3 \leqslant \Pi_1 + \Pi_2$, ceea ce se mai poate scrie

$$p(x) \leqslant p(x \vee y) \leqslant p(x) + p(y),$$

$$p(y) \leqslant p(x \vee y) \leqslant p(x) + p(y),$$

rezultat care este cunoscut din T3 și T5 (v. 10.2).

Formula matriceală utilizată se generalizează pentru cazul a n funcții de m variabile, care formează o bază:

$$V \cdot P = \Pi,$$

unde matricea V are ca linii valorile de adevăr ale celor m funcții, P este vectorul coloană al probabilităților lumilor posibile, iar Π vectorul coloană al probabilității funcțiilor din bază.

Tot sub formă matriceală se reprezintă dezvoltarea unei funcții $g(x_1, \ldots, x_m)$:

$$g(x_1, \ldots, x_m) = \sum_{j=1}^{2^m} d_j m_j = DM,$$

unde D este vectorul linie reprezentînd valorile de adevăr ale funcției în toate lumile posibile iar M vectorul coloană al minitermenilor. Atunci

$$p(g(x_1, \ldots, x_m)) = \sum_{j=1}^{2^m} d_j p(m_j) = DP$$

unde P este vectorul coloană al probabilităților minitermenilor (adică a lumilor posibile), întrucît $p(m_j) = p_j$ (ultima egalitate a fost definită în 10.1 pentru realizabilitate). Se mai ține seama că $\sum_{j=1}^{2^m} p(m_j) = 1$ și că în cazul unei tautologii $d_j = 1$ $(j = 1, \ldots, 2^m)$ și $p(g(x_1, \ldots, x_m))=1$.

Problema lui Boole se poate enunța în felul următor:

Fiind date $f_i(x_1, \ldots, x_m)$, unde $i = 1, \ldots, n$, cu probabilitățile cunoscute $\Pi_i = p(f_i(x_1, \ldots$ $\ldots, x_m))$, și o funcție $g(x_1, \ldots, x_m)$, să se determine marginile inferioară și superioară ale lui $p(g)$:

$$\text{MINF } g \leqslant p(g) \leqslant \text{MSUP } g.$$

Această problemă conduce la o problemă de programare liniară, în care datele cunoscute vor deveni restricții. Se știe că

$$\sum_{j=1}^{2^m} a_{ij} p_i = \Pi_i, \quad \text{unde} \quad p_j \geqslant 0 \quad (j = 1, \ldots, 2^m),$$

$$\text{și } a_{ij} = \begin{cases} 1, \text{ dacă } m_j \text{ implică } f_j, \\ 0, \text{ dacă } m_j \text{ implică } \bar{f}_j. \end{cases}$$

Folosind aceste notații și cele privind dezvoltarea unei funcții g, probema devine

$$A^f \cdot P = \Pi, \qquad p(g) = DP,$$

unde A^f este matricea valorilor de adevăr (sau a coeficienților dezvoltărilor funcțiilor f_i) completată cu vectorul unitate. Acesta ne asigură de condiția $\sum_{j=1}^{2^m} p_j = 1$.

Ca problemă de programare liniară, problema lui Boole se enunță astfel: să se găsească $\min q = DP$ și $\max q = DP$, unde q este o formă liniară, cu restricțiile liniare $A^f P = \Pi$, unde $P \geqslant 0$.

Se procedează prin rezolvarea problemelor duale folosind metoda Motzkin-Fourier [5]:

$$(A^f)^T \cdot Z \leqslant D^T \qquad (A^F)^T \cdot X \leqslant D^T$$

$$\max u = (\Pi)^T Z \qquad \min v = (\Pi)^T X$$

care conduc la soluțiile $z_1, \ldots, z^r$ și $x_1, \ldots, x^s$.

Scanned with OKEN Scanner

Marginile căutate vor fi:

$$\text{MINF } p(g) = \max [(\Pi)^T z_1, \ldots, (\Pi)^T z^r],$$

$$\text{MSUP } p(g) = \min [(\Pi)^T x_1, \ldots, \Pi)^T x^s].$$

Exemplificăm aplicarea metodei în cazul $f_1 = P_2$, $f_2 = P \to Q$ și $g = Q$. În acest caz:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \quad AP = \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix}$$

$$\Pi_1 = p_1 + p_2 = p(f_1) = p(P)$$

$$\Pi_2 = p_1 + p_3 + p_4 = p(f_2) = p(P \to Q)$$

$$\Pi_3 = p(g)$$

$$p_1 + p_2 + p_3 + p_4 = 1$$

$$P \geqslant 0, \qquad D = [1 \quad 0 \quad 1 \quad 0], \qquad DP = p_1 + p_3.$$

Problema de programare liniară devine: să se găsească $\max q = p_1 + p_3$ și $\min q = p_1 + p_3$, cînd $AP = \Pi$. Problemele duale sînt:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \geqslant \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{și} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \leqslant \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\min(\Pi_1 x_1 + \Pi_2 x_2 + x_3) \qquad \max(\Pi_1 z_1 + \Pi_2 z_2 + z_3)$$

$$x_1 + x_2 + x_3 \geqslant 1 \qquad z_1 + z_2 + z_3 \leqslant 1$$

$$x_1 + x_3 \geqslant 0 \qquad z_1 + z_3 \leqslant 0$$

$$x_2 + x_3 \geqslant 1 \qquad z_2 + z_3 \leqslant 1$$

$$x_2 + x_3 \geqslant 0 \qquad z_2 + z_3 \leqslant 0$$

Soluțiile sînt $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ și $z_1 = 1$, $z_2 = 1$, $z_3 = -1$ Prin urmare, $\min (\Pi_1 x_1 + \Pi_2 x_2 + x_3) = \Pi_2$ și $\max(\Pi_1 z_1 + \Pi_2 z_2 + \Pi_3 z_3) = \Pi_1 + \Pi_2 - 1$. În final

$$\Pi_1 + \Pi_2 - 1 \leqslant \Pi_3 \leqslant \Pi_2.$$

Am regăsit pe această cale teorema T10 (v.10.2):

$$p(P) + p(P \to Q) - 1 \leqslant p(Q) \leqslant p(P \to Q).$$

Există și o altă cale de a rezolva problema și anume, formulînd criteriul maximizării entropiei:

$$H = -\sum_{i=1}^{n} p_i \log p_i = -(p)^T |\log P|.$$

Se recurge la multiplicatorii lui Lagrange și în cazul particular, tratat mai înainte. Soluția este

$$P = \begin{bmatrix} \Pi_1 + \Pi_2 - 1 \\ 1 - \Pi_2 \\ (1 - \Pi_1)/2 \\ (1 - \Pi_1)/2 \end{bmatrix},$$

Scanned with OKEN Scanner

ceea ce conduce la

$$p(Q)=\Pi_3=[1,\ 0,\ 1\ \ 0].\quad P=\frac{1}{2}\,\Pi_1+\Pi_2-\frac{1}{2}=\frac{1}{2}\,p(P)+p(P\to Q)-\frac{1}{2}$$

Nu mai obținem marginile pentru $p(Q)$, ci o valoare aproximativă care se înscrie în marginile stabilite pentru $p(Q)$ prin programarea liniară.

Comentarii. Problema lui Boole a reapărut în zilele noastre în sistemele de inteligență artificială (ca de exemplu sistemele expert). Să presupunem că într-o bază de cunoștințe cu diferite grade de certitudine vrem să includem o cunoștință nouă. Pentru ca baza să rămână consistentă gradul de certitudine al noii cunoștințe nu poate fi arbitrar, el trebuie să se încadreze în anumite limite determinate de cunoștințele existente. Rezolvarea directă a acestei probleme fie prin programarea liniară, fie prin multiplicatorii lui Lagrange este laborioasă așa cum rezultă din 10.3.

În multe cazuri practice în care matricele A și V sînt de dimensiuni mici, se poate recurge la procedee ad-hoc și la soluții aproximative [11].

10.4. PROBABILITATEA CONDIȚIONATĂ ȘI CONDIȚIONALUL LOGIC

În calculul probabilităților, probabilitatea condiționată și formula lui Bayes s-au dovedit a fi un puternic instrument de modelare matematică, în special a proceselor de decizie. Atribuim unei ipoteze H probabilitatea apriori $p(H)$. În urma unui nou eveniment numit evidență E, probabilitatea lui H în condițiile producerii lui E, $p(H|E)$ este probabilitatea condiționată a lui H. Ea se definește ca $p(H|E)=\dfrac{p(H\cdot E)}{p(E)}$. Formula $p(H|E)\cdot p(E)=p(E|H)\cdot p(H)=p(E\cdot H)$, unde $E\cdot H=E\wedge H$ este realizarea împreună a evidenței și ipotezei, ne conduce la formula lui Bayes.

$$p(H|E)=p(H)\cdot\frac{p(E|H)}{p(E)},$$

unde $p(H|E)$ este o probabilitate aposteriori. Aceasta permite aducerea la zi a probabilităților atribuite unei ipoteze, pe măsura apariției de noi evidențe.

În cazul general, în prezența mai multor evidențe mutual exclusive și exhaustive $E_1,\ldots,E_n$ sînt valabile formulele

$$p(H)=\sum_{j=1}^{n}p(H\cdot E_j),$$

$$p(H\cdot E_i)=p(H|E_i)\cdot p(E_i)),\qquad i=1,\ldots,n,$$

$$p(H)=\sum_{j=1}^{n}p(H|E_j)\cdot p(E_j)$$

$$p(E_i|H)=p(E_i)\,\frac{p(H|E_i)}{\sum_{j=1}^{n}p(H|E_j)\cdot p(E_j)},\qquad i=1,\ldots,n.$$

Ultima formulă se numește *formula generalizată a lui Bayes.*

Scanned with OKEN Scanner

Folosirea probabilității condiționate în logică unde formulele iau locul evenimenelor este susceptibilă de a crea confuzii. Notația $H|E$ se citește atît „H cînd E" cît și „dacă E, atunci H" ca și implicația. Faptul că bara $|$ indică o legătură între H și E cu totul diferită de implicație o arată relația T12 (v. 10.2)

$$p(E \to H) = p(EH) - p(E) + 1,$$

din care se obțin relațiile „barei condiționale" cu implicația :

$$p(H|E) = \frac{p(E \to H) + p(E) - 1}{p(E)},$$

$$p(E \to H) = p(H|E) \cdot p(E) + 1 - p(E).$$

Exemplu. Fie următoarele două propoziții într-un joc de zaruri :

A : „rezultatul (aruncării zarului) va fi un număr mai mic ca 3",

B : „rezultatul va fi un număr par (2, 4, 6)".

Avem

$$p(A) = \frac{2}{6} = \frac{1}{3}; \quad p(B) = \frac{1}{2}; \quad p(AB) = \frac{1}{6}, \quad p(B|A) = \frac{p(AB)}{p(A)} = \frac{1}{2}.$$

Întocmim un tabel complet al rezultatelor posibile și calculăm implicația $A \to B$:

	B	1	2	3	4	5	6
	A	0	1	0	1	0	1
1	1	0	1	0	1	0	1
2	1	0	1	0	1	0	1
3	0	1	1	1	1	1	1
4	0	1	1	1	1	1	1
5	0	1	1	1	1	1	1
6	0	1	1	1	1	1	1

Implicația $A \to B$ este adevărată în 30 cazuri din 36 :

$$p(A \to B) = \frac{5}{6},$$

ceea ce verifică formula

$$p(A \to B) = p(B|A)\,p(A) + p(\bar{A}) = \frac{1}{2} \cdot \frac{1}{3} + \frac{2}{3} = \frac{5}{6}.$$

Se observă în acest tabel că pentru calculul lui $p(B|A)$ au fost luate în considerare numai cele două linii de sus, în care A era adevărată, în timp ce pentru $p(A \to B)$ au fost luate în considerare și cazurile în care A este falsă, cazuri în care $A \to B$ este adevărată, indiferent de valoarea de adevăr a lui B.

Vom nota pe $B|A$ cu semnul logic $A > B$ (a nu se confunda cu relația $<$ dintre probabilități). Vom numi pe $>$ *condițional* și vom citi $A > B$: „A condiționează pe B". (Unii autori numesc implicația „condițional material", iar condiționalul „condițional indicativ".) În consecință,

$$p(A > B) = \frac{p(AB)}{p(A)}.$$

Scanned with OKEN Scanner

Tabla de adevăr pentru implicația materială $\rightarrow$, implicația strictă $\Rightarrow$ și condițional $>$ relevă diferențele esențiale între acestea :

A	B	$A \rightarrow B$	$A \Rightarrow B$	$A > B$
1	1	1	1	1
1	0	0	—	0
0	1	1	1	—
0	0	1	1	—

A spune că $A \rightarrow B$ este o tautologie sau că are loc implicația strictă $A \Rightarrow B$ revine la omiterea liniei a doua din tablă. A spune că $A > B$ revine la omiterea ultimelor două linii, corespunzătoare cazului $A=0$, pe care formula $p(A > B) = \dfrac{p(AB)}{p(A)}$ îl exclude.

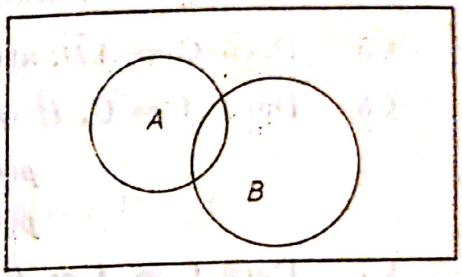


Fig. 10.2

Reprezentarea grafică din fig. 10.2 ilustrează pe $A > B$ (a se compara cu fig. 10.1 de la 10.2).

Pentru comparația condiționalului cu implicația sînt relevante următoarele relații și proprietăți :

a) $AB \Rightarrow A > B \Rightarrow A \rightarrow B$.
Se verifică ușor că $p(AB) \leqslant p(B|A) \leqslant p(A \rightarrow B)$.

b) *Contrapunerea*. Pentru implicație are loc relația

$$A \rightarrow B \Leftrightarrow \bar{B} \rightarrow \bar{A}$$

care nu este valabilă în cazul condiționalului. Dacă $A > B \Leftrightarrow \bar{B} > \bar{A}$, atunci $p(A > B) = p(\bar{B} > \bar{A})$, ceea ce ar presupune $p(AB) = p(A)$, care nu are loc în cazul general.

c) *Tranzitivitatea* este valabilă pentru implicație :

$$(A \rightarrow B)(B \rightarrow C) \Rightarrow (A \rightarrow C),$$

dar nu și pentru condițional.

Se observă că $p((A > B)(B > C)) \leqslant p(A > C)$ nu este adevărată pentru $AB \neq 0$, $BC \neq 0$, $AC = 0$.

d) *Monotonia* este valabilă pentru implicație : din $A \rightarrow C$ rezultă $AB \rightarrow C$, oricare ar fi B. Dar din $A > C$ nu putem infera $AB > C$.

Este suficient să presupunem $p(A > C) \leqslant p(AB > C)$ sau

$$\frac{p(AC)}{p(A)} \leqslant \frac{p(BC) + p(A) - p(BC \vee A)}{p(AB)}$$

să observăm că atunci cînd $BC = 0$, $AC = 0$, contrar ipotezei.

e) Relațiile $A \rightarrow 1$, $A \rightarrow A$ sînt valabile și pentru condițional : $A > 1$, $A > A$.

Din definiția probabilității condiționate și din definiția condiționalului

$$p(A > B) = p(B|A) = \frac{p(AB)}{p(A)}$$

deducem cu ajutorul proprietăţilor P1—P4 (v.10.1) şi T1 — T12 (v. 10.2) următoarele proprietăţi ale condiţionalului :

C1 $p(A > B) \geqslant 0$.

C2 Dacă $A \Rightarrow B$, $p(A > B) = 1$.

C3 Dacă $A \Rightarrow B \to C$, atunci $p(C > A) \leqslant p(C > B)$.

Demonstraţie. Din $A \Rightarrow B \to C$ urmează $A \to B \Rightarrow A \to C$ şi $p(A \to C) \leqslant p(A \to B)$. Dar $p(A \to C) = p(AC) - p(A) + 1$ şi $p(A \to B) = p(AB) - p(A) + 1$, de unde $p(AC) \leqslant p(AB)$ şi $p(C > A) \leqslant p(C > B)$.

C4 Dacă $B \Leftrightarrow C$, atunci $p(B > A) = p(C > A)$.

C5 Dacă $C \Rightarrow \overline{AB}$, atunci $p(C > A \vee B) = p(C > A) + p(C > B)$.

C6 Dacă $A \Rightarrow C$, $B \Rightarrow C$, $A \Rightarrow D$, $B \Rightarrow D$, atunci

$$\frac{p(C > A)}{p(C > B)} = \frac{p(D > A)}{p(D > B)}.$$

C7 Dacă $C \Rightarrow A \Leftrightarrow B$, atunci $p(C > A) = p(C > B)$.

C8 $p(C > AB) = p(C > A) \cdot p(AC > B)$.

C9 $$p(BC > A) = p(C > A) \cdot \frac{p(AC > B)}{p(C > B)}.$$

C10 $$\frac{p(BC > A_1)}{p(BC > A_2)} = \frac{p(C > A_1)}{p(C > A_2)} \cdot \frac{p(A_1C > B)}{p(A_2C > B)}.$$

10.5. INFERENŢE INDUCTIVE

Spre deosebire de inferenţa deductivă care operează de la general la particular, inferenţa inductivă operează de la particular la general. Cunoştinţele particulare sînt incomplete, dar se presupune că sînt suficient de certe pentru a sta la baza unei inferenţe. Condiţionalul joacă în inferenţa inductivă un rol analog cu acela al implicaţiei stricte în inferenţa deductivă.

În prezentarea unor reguli de inducţie, vom desemna prin E o evidenţă sau o cunoştinţă de care dispunem, iar gradul de certitudine a condiţionării de către E a enunţului A va fi, conform definiţiilor introduse, $p(E > A)$. Ştiind că există o condiţionare $E > A$, ce putem spune despre gradul de certitudine al lui $E > C$, unde C este o concluzie a lui E şi A? Răspunsul este dat de următoarele reguli stabilite pe baza proprietăţilor C1 — C10 (v. 10.4) :

I_1 : $E, A, B \models C$

$$\frac{p(E > A) \geqslant 1 - \varepsilon_1,\ p(E > B) \geqslant 1 - \varepsilon_2}{p(E > C) \geqslant 1 - (\varepsilon_1 + \varepsilon_2)}$$

Această regulă arată că dacă din E, A, B rezultă logic C şi dacă gradul de certitudine al condiţionării lui A şi B de către C este mai mare de un anumit prag $1 - \varepsilon$, atunci şi gradul de certitudine al condiţionării $E > C$ se situează deasupra unui prag similar.

Scanned with OKEN Scanner

Demonstrație.

a) $E \models AB \to C$ — din premisă

b) $p(E > AB) \leqslant p(E > C)$ — din C3

c) $p(E > AB) \geqslant p(E > A) + p(E > B) - 1$ — din C1–C4

d) $p(E > A) + p(E > B) - 1 \geqslant 1 - \varepsilon_1 + 1 - \varepsilon_2 - 1 = 1 - (\varepsilon_1 + \varepsilon_2)$

e) $p(E, AB) \geqslant 1 - (\varepsilon_1 + \varepsilon_2)$

f) $p(E > C) \geqslant 1 - (\varepsilon_1 + \varepsilon_2)$ — din b) și e)

I2 $$\frac{p(E > A) \geqslant 1 - \varepsilon_1,\ p(E > A \to C) \geqslant 1 - \varepsilon_2}{p(E > C) \geqslant 1 - (\varepsilon_1 + \varepsilon_2)}$$

I3 $$\frac{E, A \models C}{p(EC > A) \geqslant p(E > A)}$$

I4 $$\frac{E, A \models C,\ p(E > C) < 1,\ p(E > A) > 0}{p(EC > A) > p(E > A)}$$

I5 $$\frac{E, A \models C,\ p(E\bar{A} > C) < 1,\ 0 < p(E > A) < 1}{p(EC > A) > p(E > A)}$$

I6 $$\frac{0 < p(E\bar{A} > C) < p(EA > C),\ 0 < p(E > A) < 1}{p(EC > A) > p(E > A)}$$

Citirea lui I6 se face astfel : dacă probabilitatea condiționării lui C de către E și non A este mai mică decît probabilitatea condiționării lui C de către E și A, atunci (în condițiile enunțate la numărător) probabilitatea condiționării lui A de către E este mai mică decît probabilitatea condiționării lui A de către E împreună cu C.

Inferența inductivă prezentă în proporții dominante în raționamentul uman curent, caracteristică investiției științifice exprimentale și identificată în cercetarea teoretică, este obiectul unor intense preocupări determinate de scopurile inteligenței artificiale.

10.6. LOGICA PROBABILISTĂ ȘI LOGICA MODALĂ

Pentru a stabili o corespondență între logica propozițiilor probabilistă și logica modală, vom introduce modalitățile prin proprietățile probabiliste ale formulelor stipulate în axiome.

Dacă A este o tautologie, ceea ce în scriere modală devine $\Box A$, $p(A)=1$ și deci $p(\Box A) = 1$.

Dacă A este nerealizabilă, $p(A) = 0$.

Dacă A este posibilă (realizabilă), atunci $p(A) \neq 0$, ceea ce corespunde la $\Diamond A$.

Valorile de adevăr și valorile probabiliste ale propozițiilor modale vor fi astfel următoarele :

	Valoarea de adevăr	*Valoarea probabilistă*
$\Box A$	1, dacă A este tautologie 0 altfel	1, dacă $p(A) = 1$ 0 altfel
$\Diamond A$	0, dacă A este nerealizabilă 1 altfel	0, dacă $p(A) = 0$ 1 altfel

Scanned with OKEN Scanner

Spre deosebire de A, a cărei probabilitate poate fi orice număr cuprins între 0 și 1, probabilitatea formulelor $\Box A$ și $\Diamond A$ nu poate fi decît 0 sau 1.

Implicația strictă (modală) are definiția consacrată

$$A \prec B = \Box(A \to B).$$

Folosind criteriul probabilist al tautologiei, putem demonstra axiomele logicii modale.

Să considerăm A0, A2 și A5 din cap. 8, primele două caracterizînd sistemul modal **T**, iar A5 fiind definitorie pentru sistemul modal **S5** (v. 8.4).

A2 $\quad \Box A \to A.$

Demonstrație. Trebuie să arătăm că probabilitatea acestei formule este egală cu 1. De fapt A5 enunță $\Box A \prec A$ sau $\Box(\Box A \to A)$. Scrisă ca axiomă, ne-am dispensat de notația implicației modale (v. 8). Dar $p(\Box(\Box A \to A))$ este 1 dacă $p(\Box A \to A) = 1$ sau $q = p(\sim\Box A \vee A) = 1$.

Se știe că 1) $q \geqslant p(\sim\Box A) = 1 - p(\Box A)$,

2) $q \geqslant p(A)$

din T5 (v.10.2). Sînt două cazuri: dacă $p(A) = 1$, atunci $q = 1$ conform cu 2); dacă $p(A) \neq 1$, $p(\Box A) = 0$ și $q = 1$, conform cu 1).

A0 $\quad \Box(A \to B) \to (\Box A \to \Box B).$

Demonstrație. Să arătăm că

$q = p(\sim(\Box A \to B) \vee \sim\Box A \vee \Box B) = 1$. Avem în virtutea lui T5 (v. 10.2):

1) $q \geqslant p(\sim\Box(A \to B)) = 1 - p(\Box(A \to B))$,

2) $q \geqslant p(\sim\Box A) = 1 - p(\Box A)$,

3) $q \geqslant p(\Box B)$.

Conform cu 1), $q = 1$, în afara cazului $p(\Box(A \to B)) = 1$. Recurgem la 2) și $q = 1$ în afara cazului $p(\Box A) = 1$. Dar din $p(\Box(A \to B)) = 1$ și $p(\Box A) = 1$ rezultă prin MP: $\Box(A \to B)$, $\Box A \Rightarrow \Box B$ și deci $p(\Box B) = 1$. Astfel $q = 1$ în virtutea lui 3).

A5 $\quad \Diamond A \to \Box\Diamond A$

Demonstrație. Calculăm $q = p(\Diamond A \to \Box\Diamond A) = p(\sim\Diamond A \vee \Box\Diamond A)$.

1) $q \geqslant 1 - p(\Diamond A)$. $\quad$ 2) $q \geqslant p(\Box\Diamond A)$.

Conform cu 1) $q = 1$, în afara cazului $p(\Diamond A) = 1$. Dar atunci $p(\Box\Diamond A) = 1$ și $q = 1$ în virtutea lui 2).

O dată ce axiomele lui **S5** sînt deductibile în logica probabilistă (cu convențiile de corespondență între valorile logice și cele probabiliste enunțate mai sus), rezultă că toate teoremele din **S5** sînt valabile în logica probabilistă. Cu ajutorul teoremelor logicii probabiliste se demonstrează că regulile de inferență din **S5** conservă caracterul tautologic al formulelor în sens probabilist. Reciproc, se demonstrează că orice tautologie în sens probabilist este și o teoremă a sistemelor **S5**.

Menționăm că există și posibilitatea construirii de sisteme modale cu operatorul „probabil" fără a se recurge la calculul probabilităților. Să considerăm sistemul **S5**Π format din axiomele și regulile lui **S5** cu un operator suplimentar Πp (p este probabil), introdus de axiomele noi [4]:

$\Pi_1 \quad \Pi p \to \Box\Pi p$

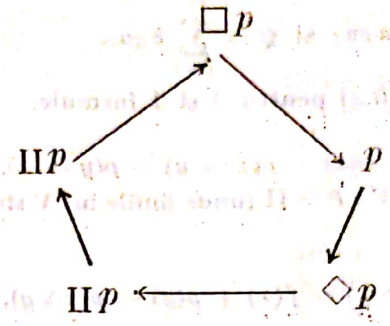
$\Pi 2 \quad \Box p \to \Pi p$

Scanned with OKEN Scanner

$\Pi 3 \quad \Pi \sim p \rightarrow \sim \Pi p$

$\Pi 4 \quad \Box(p \rightarrow q) \rightarrow (\Pi p \rightarrow \Pi q)$

Se definește de asemenea $\Pi p = \sim \Pi \sim p$ (p nu este improbabil). Legătura între operatorii modali ai acestei logici este dată de schema



Cercetările în direcția rafinării logicilor modale prin introducerea unor noi grade de certitudine, fără a se face apel la măsura probabilității, sînt de mare actualitate.

10.7. NOTĂ ISTORICĂ

Deși sînt intim legate de nașterea logicii matematice la De Morgan și Boole în secolul trecut, încercările de a fundamenta o logică probabilistă sînt de dată recentă. Cele mai importante inițiative sînt legate de R. Carnap și H. Reichenbach între 1930—1940, un deceniu de expansiune al logicilor neclasice. Raportul calculului probabilităților cu logica este discutat continuu în marea dezbatere care are loc cu privire la fundamentele calculului probabilităților și, mai ales, în confruntarea între școala „obiectivistă" care se bazează pe probabilitate ca frecvență statistică și cea „subiectivistă" care pornește de la probabilitatea subiectivă și de la formula lui Bayes. Matematicianul Kolmogorov, autorul axiomelor pentru teoria probabilităților, s-a ocupat de logica matematică. O importantă contribuție la această dezbatere teoretică a avut-o școala românească de probabilități [12].

Din anii 1960 apar noi studii importante asupra logicii probabiliste cu predicate (D. Scott, H. Gaifman etc.).

În prezent progresele inteligenței artificiale au pus la ordinea zilei aprofundarea legilor raționamentului asupra incertitudinii. Direcțiile principale sînt următoarele : a) calcule bazate pe formula Bayes ; b) utilizarea criteriilor entropice ; v) școala Dempster-Shafer care a construit procedee care generalizează metoda bayesiană ; d) logici vagi bazate pe mulțimile vagi (L. A. Zadeh) sau pe mulțimile grosiere (școala poloneză) ; metode neprobabiliste de tipul celei menționate în 10.6 ; logici noi ca de exemplu logica de precizie variabilă (Michalski-Winston) construite în jurul unor operatori noi, în unele cazuri, ternari.

Aproape toate sistemele expert ce se elaborează în prezent presupun instrumente de modelare a raționamentului în condiții de incertitudine, iar generația a 5-a de calculatoare are nevoie de progrese însemnate în direcția construirii și minuirii bazelor de cunoștințe cu diferite grade de certitudine.

10.8. EXERCIȚII ȘI PROBLEME

1. Să se demonstreze că formulele T4, T7, T8, T9 (v. 3.11) sînt tautologii, folosindu-se criteriul probabilist și formula T10 (v.10.2).

2. Să se demonstreze formula

$$p(P) \leqslant p(P \vee Q) \leqslant p(P) + p(Q),$$

folosindu-se dezvoltările $P = \sum_{j=1}^{2^m} a_j m_j$ și $Q = \sum_{j=1}^{2^m} b_j m_j$.

3. Să se generalizeze T2 (v.10.2) pentru 3 și 4 formule.

4. Să se demonstreze

$$p(y) = p(x) + p(x \to y) - p(y \to x),$$

folosind ecuația matriceală (v.10.3) $V \cdot P = \Pi$ (unde liniile lui V sînt valorile de adevăr ale funcțiilor x, $x \to y$, y, $y \to x$).

5. Să se obțină formula

$$p(x \wedge y) = p(x) + p(y) - p(x \vee y),$$

folosind ecuația matriceală $V \cdot P = \Pi$ (v. 10.3).

6. Să se demonstreze C1, C2, C4 — C10 (v. 10.4).

7. Relația $(A > B) \wedge (B > C) \not\Rightarrow (A > C)$ devine validă cînd înlocuim pe $B > C$ prin $B \Rightarrow C$ (v. 10.4).

8. Ce se întîmplă cu formula validă

$$(A \to B)(A \to C) \Rightarrow (A \to BC)$$

dacă înlocuim implicația $\to$ prin condițional $>$ (v. 10.4)?

9. Să se demonstreze I2 — I6 (v.10.5).

10. Să se arate că în logica S5 Π avem

$$\Pi(p \to Xq) \Leftrightarrow \Pi p \to Xq \Leftrightarrow X(\Pi p \to q),$$

$$\Pi(p \to Xq) \Leftrightarrow \Pi p \to Xq \Leftrightarrow X(\Pi p \to q),$$

unde $X = \Box, \Pi, \Pi, \Diamond$ (v. 10.6).

BIBLIOGRAFIE

1. Adams, E. W. *The Logic of Conditionals*. Dordrecht-Holland, Reidel, Publ., 1975.
2. Bibel, W. *Methods of Automated Reasoning*. În : Fundamentals of Artificial Intelligence, Lecture Notes in Computer Science, nr. 232, Springer, 1986.
3. Baudot, M. *Logique Inductive et Probabilité*. Paris, Armand Colin, 1972.
4. Burgess, G. *Probability Logic*. In : Journal of Symbolic Logic, 34, nr. 2, iunie, 1969.
5. Dantzig, G. B. *Linear Programming and Extensions*. Princeton, Princeton University Press, 1963.
6. Hailperin, T. *Boole's Logic and Probability*. North Holland, 1976.
7. Hasenjaeger, G. *Introduction to the Basic Concepts and Problems of Modern Logic*. Dordrecht-Holland, Reidel, 1972.
8. Kappos, D. *Probability Algebras and Stochastic Spaces*. N. Y., Academic Press, 1969
9. Maliță Mihaela, Maliță, M. *Inferența probabilistă* (1). București, Lucrările sesiunii Institutului de filozofie, 1987.
10. Maliță Mihaela, Maliță M. *Metode probabiliste în logica propozițiilor*. În : Lucrările sesiunii Centrului de Calcul al Universității București, 1987.
11. Michalski, R. S., Winston, P. *Variable Precision Logic*. În : Artificial Intelligence, 29, aug., 1986, p. 121—146.
12. Nilsson, N. *Probabilistic Logic*. În : Artificial Intelligence, 28, 1986, p. 71—87.
13. Onicescu, O. *Principes de logiques et de philosophie mathématique*. București, Editura Academiei R.S.R., 1971.
14. Polya, G. *Descoperirea în matematică*. București, Editura Științifică, 1971.
15. Reichenbach, H. *The Theory of Probability*. University of Californis Press, 1949.
16. Rescher, N. *Topics in Philosophical Logic*. Dordrecht-Holland, Reidel Publ., 1968.
17. Shafer, G. *Mathematical Theory of Evidence*. Princeton University Press, 1976.

Scanned with OKEN Scanner

Index alfabetic

Abordare semantică 37
— sintactică 37
absorbtie 19
absurditate 16
adevăr contingent 190
adjuncție 42
adunare 16
alfabet 10, 37
— de ordinul 0 53
— de ordinul 1 53
algebra ansamblelor semnate 108
— formulelor 88
algebră 87
— B 81
— booleană 80
— — atomică 82
— — completă 92
— — liberă 87
— — polinomial completă 99
— — topologică 100, 224
— cvasibovleană 198
— ITE 149
— Lindenbaum 88
— Lukasiewicz-Moisil 196, 198
— modală 221
— — deontică 222
— — epistemică 222
— — normală 222
— — normal deontică 222
— Post 191, 194, 195
— produs 96
— pseudobooleară 79
algebre similare 87
algoritm al saturării de nivel 119
algoritmul lui Wang 111
— Quine-McCluskey 129
aplicație injectivă 57
— surjectivă 57
arbore 63
— de cvasiderivație 64
— — demonstrație 63
— — derivare 63
asociativitate 19
atom 9, 81
— al unei algebre booleene 81
axioma lui Brouwer 215
axiomă 37, 62 67
axiomele sistemului formal 37

Bază 101
bijecție 57
bloc 124
— accesibil 220

Calcul cu scheme 61
— natural 61
calculul circuitelor cu contacte 150
caracteristica unui inel 82
cel mai mare element 72
— — mic element 72
ciclare 187
circuit 156
— logic combinațional 165
— stea 156
— triunghi 156
clauză 29
— centrală 121
— conjunctivă 29, 32
— disjunctivă 29, 32
— Horn 147
— laterală 121
— l-negativă 113
— l-neutră 113
— l-pozitivă 113
— vidă 113
— virf 121
codificator 166
comparator digital 167
complement 76, 138
completitudine 47
complexitate 146
compoziție 42
comutativitate 19
condițional 252
conector 9
— adecvat 23
— al lui Nicod 23
— — — Sheffer 23
— binar 11
— unar 11
congruențe proprii 87
conjuncție 22
consecință logică 26, 67
consistență 48
contact 150
— bipozițional 150
— complementar 151
— deschis 150
— închis 150
contingentul 206
contradicție 15, 26
contrapunere 253
cuantificator 9
cvasiderivație a unui secvent 64
cvasiordine 28, 73

Decidabilitate 51
decodificator 167

Scanned with OKEN Scanner

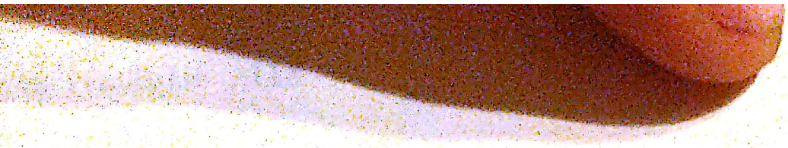
subprodus 127
substituțe prin echivalență 42
subsumare 115
sumator 168
supremum 72

Tablă de adevăr 12
tautologie 15
— intuiționistă 174
— propozițională pozitivă 180
teorema deducției 45
— recurenței 58
— substituției 66
teoremă 63
teoremă a înlocuirii 66
— de reprezentare a algebrelor boolene 108
— a idealului prim 106
teorie formalizată 67
— propozițională intuiționistă 174
transmisie 151
transpunere 20
trecere 63

Unitate 113

Variabilă de intrare 151
— metalogică 40
— propozițională 10

Scanned with OKEN Scanner



Redactor : VALENTINA CREȚU
Tehonoredactor : MARIA TRĂSNEA

Bun de tipar : 02.12.1987;
Coli tipar : 16,50
C. Z. : 681.3 :517.11

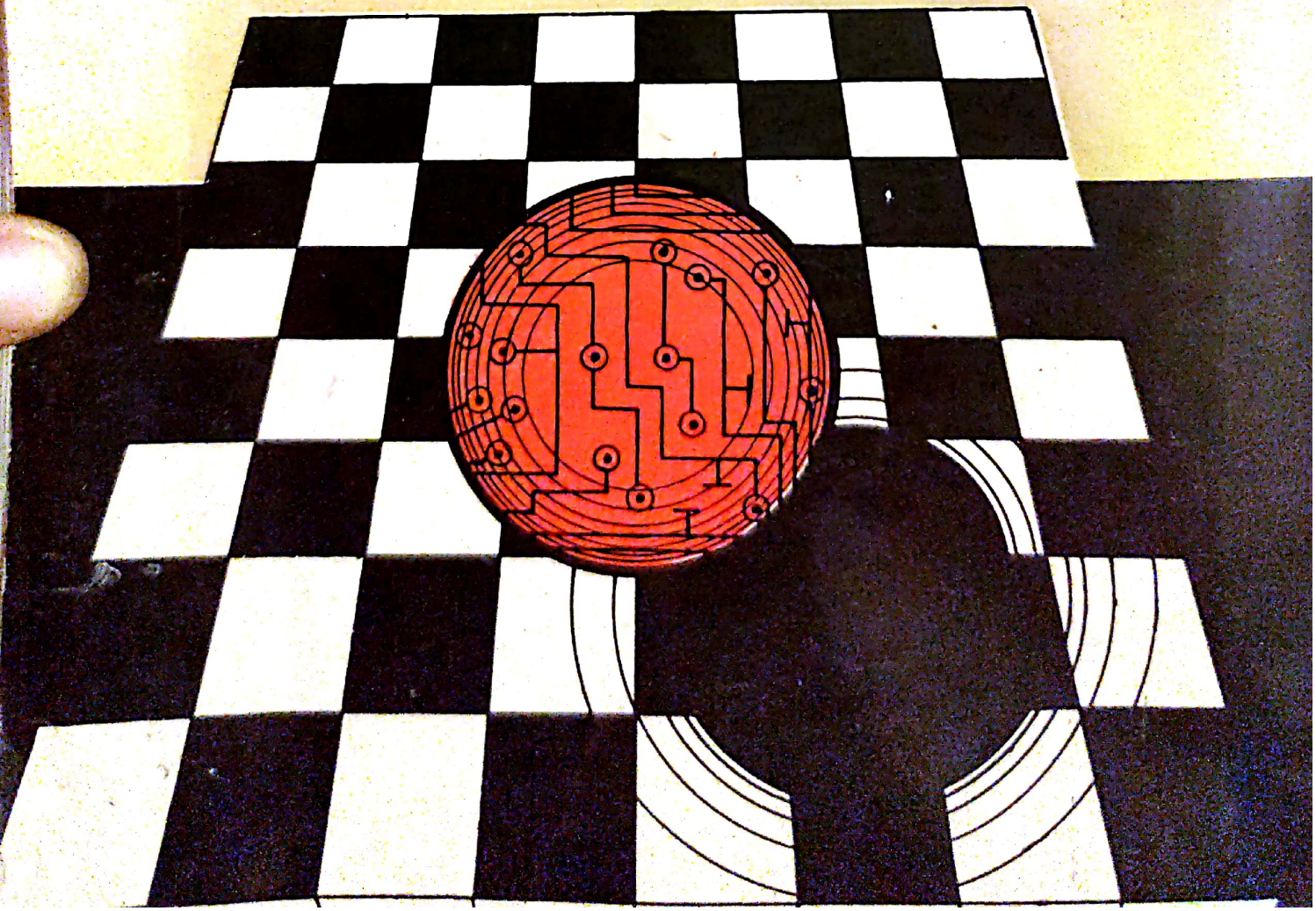
---

c. 288 I. P. Informația
str. Brezoianu nr. 23 — 25,
București

Scanned with OKEN Scanner

Lei 22,50

Editura Tehnică



Scanned with OKEN Scanner